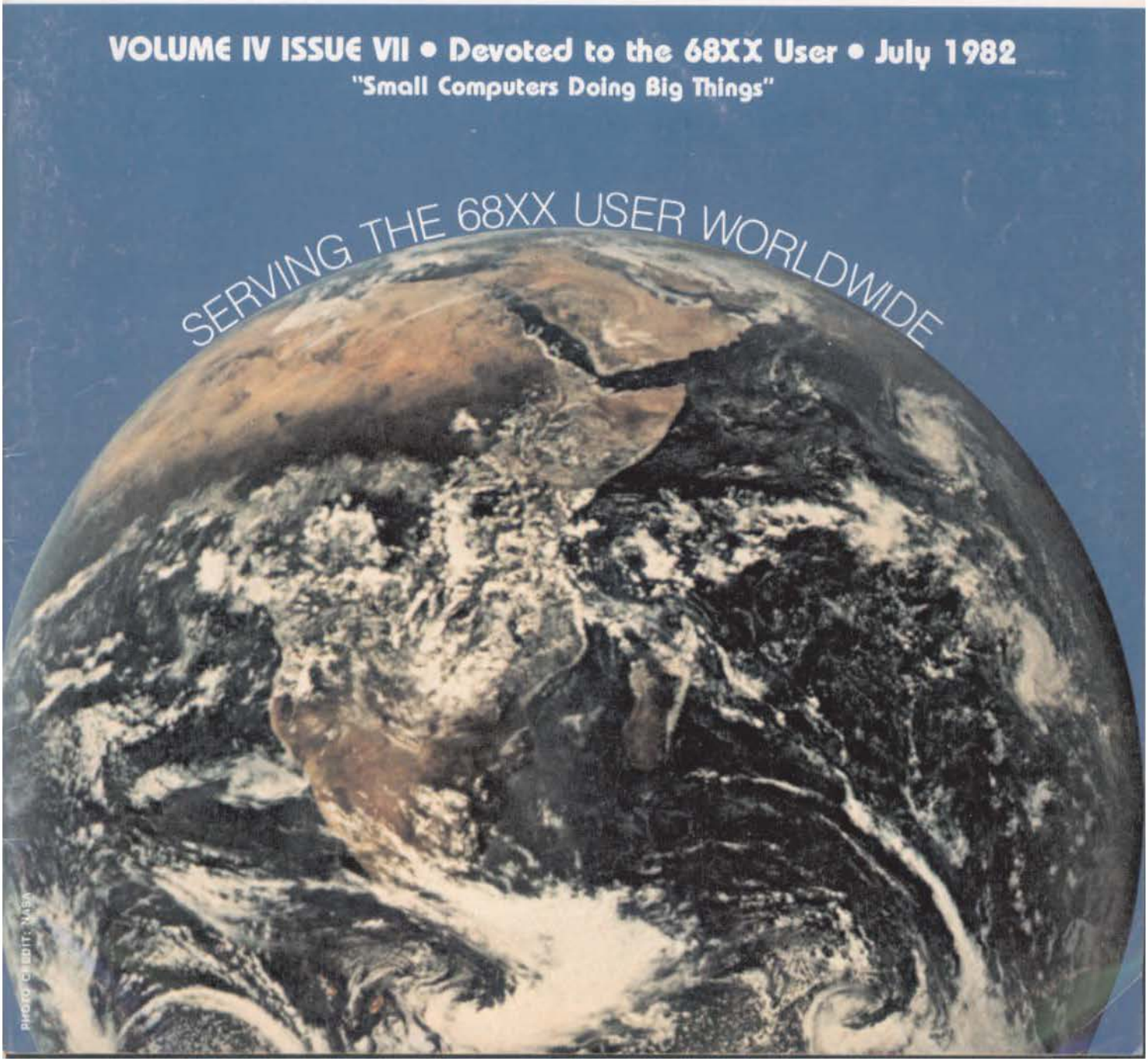
'68' MICRO JOURNAL

$2.95USA

Australia	A $ 4.75	New Zealand	NZ $ 4.75
Singapore	S $ 9.45	Hong Kong	H $23.50
Malaysia	M $ 9.45	Sweden	30:-SEK

VOLUME IV ISSUE VII • Devoted to the 68XX User • July 1982

"Small Computers Doing Big Things"



YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or **8209** — nine-inch diagnonal screen

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

Pascal for 6809

Pascal for the 6809 is a true native code compiler. Unlike the usual P-code Pascals which run in an interpretive manner, ours produces efficient assembly language mnemonics which can be assembled and run directly. This compiler is available for both 6809 FLEX™ and UniFLEX™. Many features not found in other Pascal systems were implemented while avoiding those features completely non-standard. Features of the Pascal system include:

- Supports most of Jensen and Wirth specification
- Produces fast and efficient 6809, native code
- FLEX run-time package may be trimmed
- Double precision real numbers (16.8 digits)
- Implements scalar, subrange and structured data types
- Standard I/O using file buffer pointers
- Dynamic storage allocation
- Ability to call other Pascal programs
- FLEX version may call assembly language programs
- Buffered or single character terminal input
- Standard math functions: SIN, COS, ARCTAN, EXP, LN, SQR, SQRT
- Random number generator function
- Many usable, sample programs included
- UniFLEX version supports:
  - Random file positioning
  - Ability to call various UniFLEX system routines
  - Ability to execute UniFLEX utility commands

Pascal on diskette for 5" and 8" 6809 FLEX is available for $200.00 The 5" version requires two disk drives. The UniFLEX version is $300.00 and includes one year of maintenance. All orders should include 3 percent for postage and handling (10 percent on foreign orders).

™FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.

technical systems consultants, inc.

111 Providence Rd., Chapel Hill, N.C. 27514

(919) 493-1451

'68'

MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-DMAF2-CDS1-CT82-Sprint 3
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Mainframe-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

TABLE OF CONTENTS

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
615 842-4600

68 Micro Journal is published 12 times a year by Computer Publishing, Inc. Second class postage paid at Hixson, Tennessee and additional entries. Postmaster: send Form 3579 to 68 Micro Journal, PO Box 849, Hixson, Tennessee.

Subscription Rates

1-Year $24.50 2-Years $42.50 3-Years $64.50 USA

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors **full name, address, date and telephone number.** It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands **ONLY** should be used (due to our proportional processor): .sp space, .pp paragraph, .fi fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continous text.

All articles submitted on diskettes should be in TSC FLEX™ format, either FLEX2 6800, or FLEX9 6809 any version.

If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). **Please use a dark ribbon!**

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word. 20 words or less $7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.

2MHZ 6809 SYSTEMS

GIMIX offers you a variety to choose from!

38 MB WINCHESTER SYSTEM $17,498.99

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 512KB Static RAM
- ★ 8 RS232C Serial Ports
- ★ 2 Parallel Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ Dual 8" DSDD Floppy Disk System
- ★ Dual Winchester Subsystem with Two19 MB 5¼" Winchester Drives

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Debugger
- ★ OS-9 Text Editor
- ★ OS-9 Assembler

19 MB WINCHESTER SYSTEM $8998.09

HARDWARE FEATURES:

- ★ 128K Static Ram
- ★ 2MHz 6809 CPU
- ★ 19 MB 5¼" Winchester DMA Subsystem
- ★ 4 RS232C Serial Ports
- ★ 1 MB 5¼" Floppy Disk Drive
- ★ DMA Double Density Floppy Disk Controller

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Text Editor
- ★ OS-9 Debugger
- ★ OS-9 Assembler

128KB MULTI-USER SYSTEM $6997.39

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ DMA Double Density Floppy Disk Controller
- ★ 128KB Static Ram
- ★ 2 RS232C Serial Ports
- ★ Dual 8" DSDD Floppy Disk System

SOFTWARE FEATURES: Your choice of either UniFLEX or OS-9 LEVEL TWO. Both are Unix-like Multi-User/Multi-Tasking Operating Systems.

56KB FLEX/OS-9 "SWITCHING" SYSTEM $4148.49

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 56K Static Ram
- ★ 2 RS232C Serial Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ 2 Built-in 5¼" 40tr DSDD Disk Drives (80 Track DSDD Drive Option . . add $400.00)

SOFTWARE FEATURES:

- ★ GMXBUG monitor — FLEX Disk Operating System
- ★ OS-9 LEVEL ONE Multi-tasking operating system for up to 56K of memory

WINCHESTER SUBSYSTEMS

Winchester packages are available for upgrading current **GIMIX** 6809 systems equipped with DMA controllers, at least one floppy disk drive, and running FLEX, OS-9 LEVEL ONE or OS-9 LEVEL TWO. The packages include one or two 19MB (unformatted) Winchester drives, DMA Hard Disk Interface, and the appropriate software drivers. The interface can handle two 5¼" Winchester Drives, providing Automatic Data Error Detection and Correction: up to 22 bit burst error detection and 11 bit burst error correction.

Dual drives can be used together to provide over 30 MBytes of on line storage -- or use one for back-up of the other. (More convenient and reliable than tape backup systems.

#90 includes one 19MB Drive, Interface, and Software **$4288.90**
#91 includes two 19MB Drives, Interface, and Software **$6688.91**

Contact GIMIX for systems customized to your needs or for more information.

50 HZ Export Versions Available

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

Build performance into your system using the OS-9™ toolbox!

Microware's OS-9 Toolbox holds the finest 6809 programming languages and software tools in the industry. Microware languages are complete language implementations that meet or exceed applicable industry standards, and have unmatched speed, efficiency, and ease of use that Microware software is famous for. *All Microware languages run on OS-9 Levels One or Two.*

OS-9 PASCAL™ LANGUAGE COMPILER

The ●S-9 Pascal language compiler is the most complete and versatile Pascal available for the 6809. ●S-9 Pascal has the unusual capability of generating P-code for interpretive execution while debugging OR highly optimized 6809 assembly language source code output for maximum speed. Another feature of ●S-9 Pascal is its "virtual memory" P-code interpreter that lets you run incredibly large Pascal programs. OS-9 Pascal meets the ISO 7185.1 Standard and the complete Wirth/Jensen specification.

CIS COBOL™ COMPILER

6809 CIS Cobol is a compact, interactive and standard Cobol language compiler which is ideal for the most demanding business applications. Standard features are: ISAM, Debug, ACCEPT/DISPLAY, and Interprogram Communications modules. CIS Cobol is *the* preeminent microcomputer Cobol in the industry, and the ●S-9 version retains full compatibility with CP/M applications software. CIS Cobol meets the ANSI 1974 Level One C●BOL standard and is CSA certified. Also available is Micro Focus' FORMS 2, an optional automatic program generator that lets you interactively design screen-oriented applications with ease.

BASIC●9™ STRUCTURED BASIC INTERACTIVE COMPILER

Basic●9 is the fastest and most comprehensive full Basic language available for the 6809. It combines standard Basic with the best features of Pascal. It is a unique interactive compiler that combines compiler speed, interpreter friendliness, and superlative debugging facilities. RunB, a ROMable run-time system for compiled Basic●9 programs is now available as an option.

C LANGUAGE COMPILER

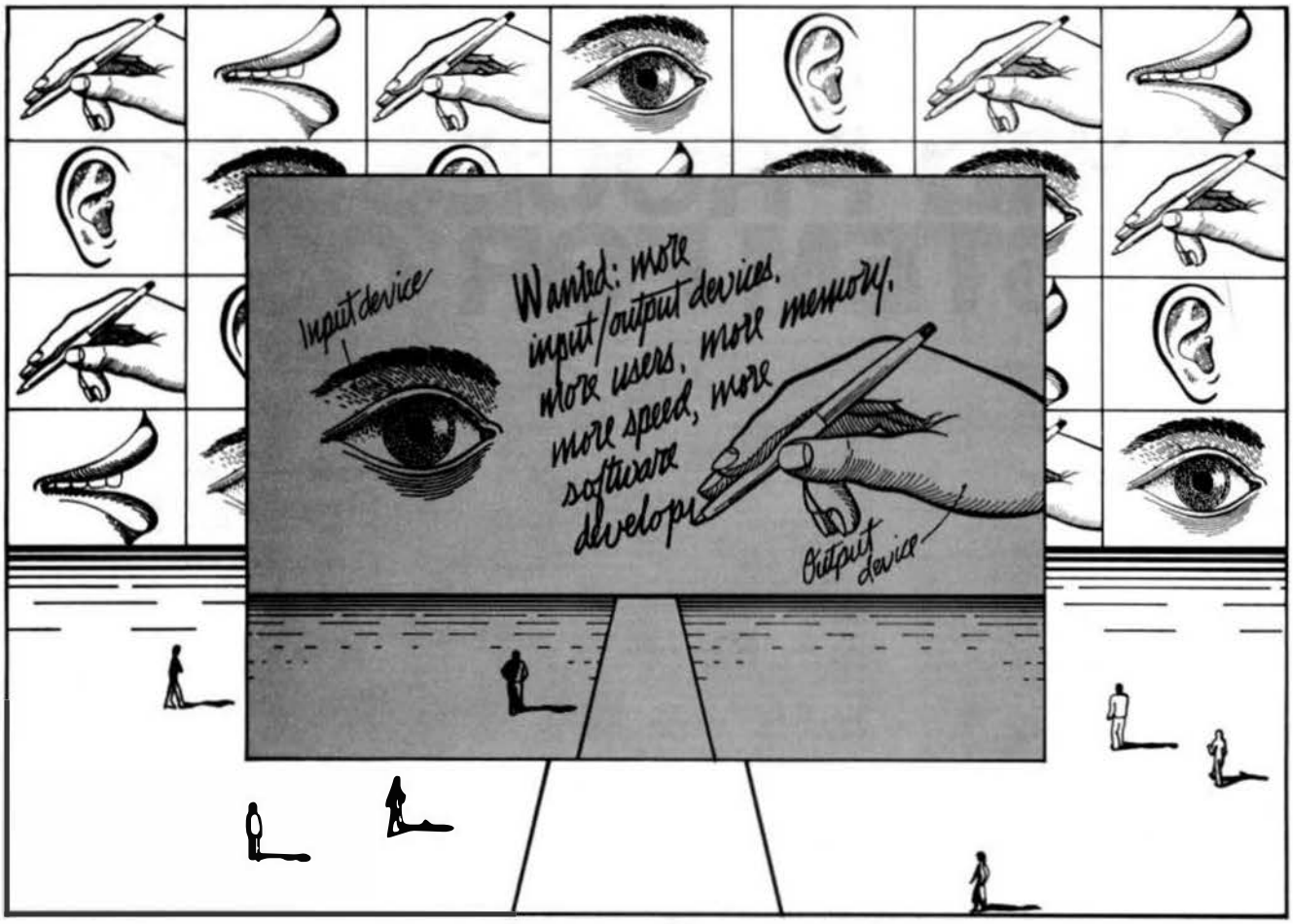
C—the systems language of the future—is here today on OS-9. This is a complete implementation of the Unix Version 7 C language including INT, CHAR, SIGNED, UNSIGNED, FLOAT and LONG data types, structures, unions, standard C library, and a full preprocessor with macro definitions. Generates fully reentrant 6809 assembly language source code output.

For information contact your computer supplier, or

MICROWARE®

Microware Systems Corporation
5835 Grand Ave., Des Moines, Ia. 50312
Ph. 515-279-8844 • Telex 910-520-2535

*Unix is a trademark of Bell Laboratories. CIS Cobol is a trademark of Micro Focus, Inc. OS-9 and Basic●9 are trademarks of Microware and Motorola, Inc.



Expand your horizons with OS-9™ Level Two

Expand your 6809 computer to a fast, efficient multi-user system utilizing up to one megabyte of memory, almost any I/O device, and comprehensive implementations of the most-wanted programming languages: Basic, C, Pascal, Cobol, and Assembler.

With OS-9 Level Two, your computer is transformed into the ultimate software development system with performance and features found only on large and costly computing systems. It brings to your fingertips the friendliness and power of a Unix*-based environment. You can even run most Unix* software tools using OS-9's Unix Version 7 compatible C Compiler.

As a multiuser system, OS-9 Level Two excels with a multi-level directory system, a fast random access file system with record lockout, user name/password logon protection, "pipes" for interprogram communication, and full file security. The versatile "Shell" command interpreter makes it easy for each user to run interactive or multiple background programs with I/O redirection to or from any file or I/O device.

As a real-time system, your OS-9-based computer can sense, monitor, control, and communicate with the real world thanks to OS-9's highly modular and user expandable structure. Adding customized I/O to OS-9 is almost too easy: software interfaces are simple, modular, and well documented. That is why OS-9 users routinely interface it to almost every kind of peripheral device and instrumentation.

For large systems, OS-9 Level Two can handle over one megabyte of RAM and ROM memory with extraordinary efficiency. When, as is often the case, two or more tasks run the same program (such as Basic09) they automatically "share" just one copy in memory. Also, OS-9 Level Two typically resides in less than 24K memory. Savings like these give OS-9 based systems large capacity without having to resort to performance-robbing techniques such as disk swapping.

OS-9 Level Two is available exclusively from manufacturers of most popular 6809 computers equipped with memory management hardware. They offer versions specifically tailored to their computers for use with both new and existing systems.

For more information about OS-9 Level Two contact your computer supplier, or

MICROWARE®

Microware Systems Corporation
5835 Grand Avenue, Des Moines, Iowa 50312
Phone 515-279-8844 • Telex 910-520-2535

*Unix is a trademark of Bell Laboratories. OS-9 is a trademark of Microware and Motorola, Inc.

Dynastar

WORD PROCESSING SYSTEM FOR OS-9

OS-9 USERS:

If your computer has a SCREEN and you're still struggling with an editor that only knows about LINES, then obviously YOU don't know about

DynaStar

DynaStar is a powerful, menu-driven screed editor equally suited to the tasks of program preparation and document processing. With the addition of the optional DynaForm print formatter, it is the best word-processing package you can buy for your OS-9 system.

DynaStar Version II is now available and features non-non- nse "what you see is what you get" editing for virtually any terminal with or without cursor addressing (it must be at least able to go to "home"). To edit, simply place the cursor where you want it, and type. Any printable character you type is entered directly into your text, and any non-printable control character causes immediate execution of an editing command. Single keystroke commands permit movement of the cursor in any direction, by character, tab, word, line, or screen full, and deletion of characters, words (left or right) or a whole line. Two keystroke commands augment this set by moving the cursor to the left margin, top or bottom of the screen, beginning or end of the edit buffer, or the beginning of the next paragraph. You can search for any string, replace with any other, do it again, mark arbitrary blocks of text, copy, move or delete blocks, read or write to side-files, set tabs and margins, or center the current line.

DynaStar features automatic word wrap, and it can right justify text as you enter it so you will see exactly how it will look before you print it. If you later make alterations or change the margin, you can reform the text a paragraph at a time with two keystrokes. For programmers, there is a special automatic indent mode to help you write well structured code. DynaStar includes a Shell command which lets you do almost anything (including edit another file) without even losing your place in your current document, and it permits editing of large disk files in stages without forcing you to break up your files.

If you want to define more powerful commands, DynaStar includes a macro facility which lets you convert any control character to one or a string of characters of your choice. You can use this feature to create global search-and-replace commands, insert "boiler-plate," or simply re-map your keyboard. You can also provide a special "start-up string" which is automatically executed whenever you enter the editor to set up modes such as auto-justify, display a directory, define your favorite macros, or re-map the keyboard.

For complete word-processing, we offer our DynaForm text formatter which provides all the standard features such as pagination, headers and footers with page numbers, single space, double space, multiple space, bold face, **double-strike**, and underline.

DynaForm has its own macro facility with string variables, nested include files, a full merge-print capability for generating form letters and mailing lists, and it can generate an index automatically, sorted alphabetically or by page number. You can call it from DynaStar to proof-print the active edit buffer, or by itself to print a disk file while you edit another.

DynaStar costs only a little more than that time-oriented editor and it is available today. If you're still not convinced that it would be the best thing that ever happened to your video terminal, you can order our "Doubting Thomas" test pak consisting of complete documentation and a special version of DynaStar that lets you edit to your heart's content, but won't update your files. Later when your doubts melt away, you can obtain credit on the full purchase price and join the faithful who bought the whole thing in the first place.

"Doubting Thomas" test pak:	$ 49.95
DynaStar II (for the faithful):	$149.95
DynaForm text formatter:	$149.95
Both purchased together:	$275.00

Note: DynaStar Version I (no macros) will be available at the original price until May 31, and current owners may upgrade to Version II with full credit until June 30.

AVAILABLE SOON FOR FLEX 9

Spelltest

Peter Dibble Puckett

FOR OS-9 AND FLEX

SPELLTEST is the most versatile 68XX spelling checker available.

MENUS MAKE OPERATION EASY. From the menu you may: Print a list of suspect words. Check each suspect word one by one. Read your text, stopping to check suspect words. Use additional dictionaries for more thorough checking or special applications. Build an additional dictionary of newly accepted words. Write correct text file to disk.

While checking you may: Accept the suspect word. Accept and save in the dictionary. Replace with correct spelling.

Designed to be used by the layman, SPELLTEST is right at home in the office. Ease of use and speed will recover the cost in days.

22,000 word dictionary covers the first 25,000 entries in the American Heritage listing of the most common English words.

500 built in common words (and, or, the, etc.) and 300 specific to your field, filter the text and allows a large file to processed even in small computers.

PRICE $199.00

A/BASIC

Basic Compiler

For OS-9 and FLEX

If you are still programming in assembler, this is the program for you! This BASIC compiler generates pure, fast efficient 6809 machine code from easy to write BASIC source programs. Uses ultra-fast integer math, extended string functions, boolean operators and run-time operations. Output is ROMable and RUNS WITHOUT ANY RUN-TIME PACKAGE. Supports IF-THEN-ELSE structure, random access, and several improvements over the original 6800 version sold by Microware. Optimized for the 6809, A/BASIC is 8 to 10 times faster than the original 6800 version, and produces code approximately 30% smaller.

SPECIAL

CHESS program coded in A/BASIC (originally sold for $50) is included FREE on the disk in both source and object for your enjoyment. Also some utilities are included for testing and examples, all in source on the disk!

ONLY $150.00

specify OS-9 or FLEX

PLOT

Now you can have GRAPHICS added to all your programs. Just write the data out to a virtual array and call PLOT. PLOT is written in TSC XBASIC and the source is included on the disk.

INFINITE RESOLUTION GRAPHICS ON YOUR TERMINAL OR PRINTER. HISTOGRAMS, BARGRAPHS, XY PLOTS PLUS OTHERS.

IN TSC XBASIC SOURCE INCLUDED ON DISK. $44.95

TOOLKIT NO1

The Basic Programmers *Toolkit*
by Dick Bartholomew

The Basic Programmers Toolkit gives the BASIC programmer the power and flexibility never before achieved under FLEX.

PRICE $49.95 object only
$69.95 with source on disk!

TOOLKIT NO2

The Programmers *Toolkit*
by Dick Bartholomew

The Programmers Toolkit is a package of utilities and programs that extend the capabilities of FLEX to the utmost.

PRICE $49.95 object only
$69.95 with source on disk!

Dynasoft PASCAL 1.4 for OS-9

Dynasoft Pascal 1.4 includes all the features of the FLEX version 1.3 with the following enhancements: Chain, Fread, Fwrite, Seek, Open, Create, Close, Delete, Fork, Send, Wait, Sleep, Settime, Time, Getstatus, Setstatus, SetPriority, GetProcID, and JSR. This is an excellent and fast program, small enough to write utilities but powerful enough for things like DynaStar.

Object only $69.95

Add for run-time source on disk $30.00

Add for source of Dynasoft Pascal 1.4 all $125.00

CRASMB

MULTI CPU CROSS ASSEMBLER FOR 6809 FLEX
by Frank Hoffman

CRASMB is a conditional macro assembler with the capability to use different CPU overlays in order to cross assemble. These CPU overlays called *CPU PERSONALITY MODULES* (*CPM's*) can be called from a source file, thereby making it easy to create object code for a variety of CPU's. It is also possible to create new *CPM's* yourself for any 8 or 16 bit CPU. The information needed is included in the manual. If you decide to do this, it would be advisable to purchase the source for one of the *CPM's* and modify it rather than starting from scratch. *CPM's* are currently available for the following CPU's: 6809, 6800, 6805, 6502, Z80/8080, 1802, and others coming.

PRICE $139.95

Includes one 8 bit *CPM* of your choice (not source). Additional *CPM's*:

8 Bit $25.00 *Source $25.00 extra*

THE BILL PAYER SYSTEM™

THE BILL PAYER is a package of 10 menu driven programs in TSC Extended Basic. This powerful system helps you keep track of your bills. You can create a vendor list, enter invoices to be paid, generate reports about them, print your checks and much more. Uses random access files.

Explore Package now included at the same price.

THE PURCHASE ORDER system adds purchase orders to the BILL PAYER. This package of programs adds another level of control to your expenditures. Prints out purchase orders and keeps track of purchases. Requires the Bill Payer to work.

INCOME/EXPENSE LEDGER. This valuable package is most appreciated at tax time. Allows up to 99 income and expense numbers. Tags into the PURCHASE ORDER system, and the Bill Payer.

Includes manual and source supplied on disk in TSC Extended Basic.

THE BILL PAYER
PURCHASE ORDER
INC/EXP. LEDGER

ALL FOR $169.95

COLOR COMPUTER USERS

THE POWERFUL FLEX DISK OPERATING SYSTEM WITH HUNDREDS OF SOFTWARE PACKAGES IS NOW AVAILABLE!

Now you can run FLEX, OS-9 and Radio Shack disk software on your Color Computer. If you have a 32K Color Computer with the Radio Shack disk system, all you need to do is make a trivial modification to access the hidden 32K, as described in the Feb. issue of COLOR COMPUTER NEWS and the April issue of '68' Micro. You can get FLEX from us right now. OS-9 will be ready by summer. Please note that this will only work with the Radio Shack disk system and 32K/64K memory chips that RS calls 32K. Maybe they put 64K's in yours, too. If you don't have a copy of the article, send a legal size SASE (40¢ stamps) and we'll send it to you.

Using this system to run FLEX and OS-9 has many advantages. First, it gives you 48K from zero right up to FLEX. This means that ALL FLEX compatible software will run with *NO MODIFICATIONS* and *NO PATCHES!* There are no memory conflicts because we moved the screen up above FLEX which leaves the lower 48K free for user programs.

What you end up with is 48K for user programs, 8K for FLEX and another 8K above FLEX for the screens and stuff. We have a multi-screen format so you can page back and to see what scrolled by and a Hi-Res screen that will enable us to have 24 lines by 42 character display is on the way. That's better than an Apple!

We also implemented a full function keyboard, with a control key and escape key. All ASCII codes can now be generated from the Color Computer keyboard!

We also added some bells and whistles to Radio Shack's Disk system when you're running FLEX or OS-9. We are supporting single or double sided, single or double density, 35, 40 and 80 track drives. If you use double sided drives, the maximum is three drives because we use the drive 3 select for side select. When you are running the Radio Shack disk it will work with the double sided drives but it will only use one side and only 35 tracks. Using 80 track drives is okay, but will not be compatible with standard Radio Shack software. You can also set each drive's stepping rate and drive type. (SS or DS - SD or DD)

In case you don't understand how this works, I'll give you a brief explanation. The Color Computer was designed so that the roms in the system could be turned off under software control. In a normal Color Computer this would only make it go away. However, if you put a program in memory to do something first (like boot in FLEX or OS-9), when you turn off the roms, you will have a full *64K RAM* System with which to run your program.

Now, we need the other half of the 64K ram chips to work, and this seems to be the case most of the time, as the article states. Of course, you could also put 64K chips in.

Some neat utilities are included.

MOVEROM moves Color Basic from ROM to RAM. Because it's moved to RAM you can not only access it from FLEX, you can run it and even change it!! You can load Color Computer cassette software and save it to FLEX disk. Single Drive Copy, Format and Setup commands plus an online help system are included.

Installing FLEX is simple. Insert the disk and type:

RUN "FLEX"

That's all there is to it! You are now up and running in the most popular disk operating system for the 6809. There are hundreds of software packages now running under the FLEX system. Open your Color Computer to a whole new world of software with FLEX.

FLEX $89.00
INCLUDES OVER 25 UTILITIES!

Other languages available include; FORTH, Pascal, Fortran77, 'C,' A/BASIC compiler, plus more.
Application packages include; A/R, G/L, A/P, Inventory, Electronic Spreadsheets, Accounting, Database programs and more. SEND FOR LIST.

TRS-80 COLOR COMPUTER COMPLETE WITH 64K RAM, 24K ROM, SINGLE DISK DRIVE AND FLEX SET UP AND READY TO RUN FOR ONLY $1,275. Includes 60 day extended warranty. If you have a Computer, call about RS disk controllers and drives.

SOFTWARE CATALOG

PROGRAM	OBJECT/WITH DISK/SOURCE		code
ILLPAYER		169.95	x
PLOT		44.95	x
TABULA RASA		100.00	x
Mailing List		99.95	x
Forms Display		49.95	x
Inventory with Material Requisition Planning		100.00	x
Some Common BASIC Programs		69.95	x
FLEX For Color Computer	99.00		9
X-FORTH (FLEX)	149.95		8 & 9
C-FORTH (TRS-80 Color)	99.95		9
TOOLKIT #1 (BASIC)	49.95/	69.95	9
TOOLKIT #2	49.95/	9.95	9
AUTOTASK		129.95	9
A/BASIC Compiler	150.00		9
Extended Utilities	49.95/	9.95	9
Pas word Protection	69.95/	89.95	9
CRASMB IX Assembler)	139.95		9
Personality Module (1 INC) 6502, 6800, 6805, 6809, Z80, 8080, 1802	25.00	50.00	9 ea
READTAPE		5e.95	9
SPELLTEST	199.00/	299.00	9
READTEST	54.95/	74.95	8 & 9
ESTHER	39.95/	59.95	8 & 9
HELP	29.95/	49.95	8 & 9
Job Control Program	49.95/	89.95	8 & 9
DYNASOFT PASCAL (FLEX)	59.95/	89.95	9
DYNASFT PASCAL (OS-9)	69.95/	99.95	9
DYNASOFT Compiler Source		125.00	P
DYNASTAR S reen Editor (OS-9)	149.95		9
SUPER SLEUTH (6800/6809)		99.00	8 & 9
SUPER SLEUTH (Z80)		99.00	8 & 9
CROSS Assembler Macros for TSC ASMB 6800/1, 6805, 6502, Z80, 8080/5	3 FOR	49.95 / 99.95	each
6502 Translator		75.00	9
Debugging Simulators 6805 or 6502		75.00	9 ea
STYLOGRAPH 2.0	295.00		9
STYLOGRAPH MAIL MERGE	125.00		9
STYLOGRAPH Spelling Checker	145.00		9

CODE X = XBASIC, 9 = 6809, 8 = 6800, P = PASCAL

Software by Technical Systems Consultants, Inc.

Flex™ (includes Editor & Assembler)	150.00
UniFLEX™ (includes one year maintenance and update)	450.00
Editor	50.00
Assembler	50.00
68000 Cross Assembler on 6809	250.00
Text Processor	75.00
Extended Basic	100.00
Basic Precompiler (specify standard or extended)	50.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™) (Add $75.00 for one year's maintenance and update)	225.00
Soft/Merge Package	75.00
6809 Flex™ Utilities	75.00
Debug Package	75.00
Diagnostic Package	75.00

Software by Microware Systems Corp.

OS-9™ Level One Operating System	200.00
OS-9™ Level Two Operating System	500.00
BASIC09™	200.00
OS-9™ Macro Text Editor	125.00
OS-9™ Interactive Assembler	125.00
OS-9™ Interactive Debugger (Disk version)	50.00
CIS Cobol Compiler	895.00
Pascal Compiler	400.00

USA add $2.50 for standard UPS shipping & handling
Foreign orders add 20% Airmail
Specify 5" or 8" disk and 6800 or 6809
VISA—MASTER CHARGE—DINERS CLUB ACCEPTED

OUR SOFTWARE IS GIMIX COMPATIBLE

AUTOTASK

WITH MENU

AUTOTASK with MENU is a revolutionary new concept designed to overcome the problems and frustrations which confront the non-technical when using a computer. Users are greeted with a series of self-prompting interactive menus linking directly to the application. Several example menus are provided. You can create your own menus from simple text files. AUTOTASK with MENU gives you unlimited software flexibility by providing a system to coordinate multiple-application programs.

Bundle several different software packages to present a coordinated system to the user. AUTOTASK with MENU is compatible with all FLEX compatible software. It uses very little memory and is easy to learn.

PRICE $129.95

Includes source on disk!

Manual $10.00

6502 TRANSLATOR
Translator 6502 code to 6809
$75.00

INVENTORY
with MATERIAL
REQUISITION PLANNING
$100.00

SUPER SLEUTH
Disassembler for 6800/6809 or Z80
$99.00

TABULA RASA
Electronic Spreadsheet
$100.00

TRS-80 COLOR COMPUTER

FORTH FOR THE TRS-80 COLOR COMPUTER DISK SYSTEM

Trying to get control of your Color Computer?? Tired of translating HEX to decimal?? Tired of remembering where the VDG and SAM are and how to program them?? Want to write machine language code with assembly language mnemonics instead of POKE$??
Want to write programs in half the time?? Want to write lots of small pieces of code that you can put together in seconds to do BIG JOBS??? Want a language that is at least 5 to 10 times faster than BASIC??? Want to learn everything there is to know about FORTH, with the best manual on the market, including lots of examples of FORTH applications, and detailed explanations of how everything works??

CC FORTH IS THE ANSWER!!
Includes Editor, 6809 Assembler
String Functions, Disk Data File
Operations and Much Much More!

$99.95

X-FORTH

FLEX COMPATIBLE FORTH

BY Chuck Eaker, Ph.D.
X-FORTH NOTES
Supplied on one 8" disk or 2 5" disks, with a 400 + page manual.
Disk(s) have the source of everything but the core.
PRICE only $149.95 plus $2.50 S&H
Manual available separately for $49.95 plus $2.50 S&H

HARDWARE CATALOG

CRT TERMINALS

Terminal		Cash	Credit ard
TVI	gtg	588.50	605.00
TVI	912/C	709.00	728.00
TVI	920/C	758.00	779.00
TVI	925	759.00	780.00
TVI	950	978.00	1004.00
Zenith	Z19	700.00	719.00
SOROC	IQ130	589.00	605.00
SOROC	IQ135	765.00	785.00
SOROC	IQ135G	831.00	853.00

PRINTERS

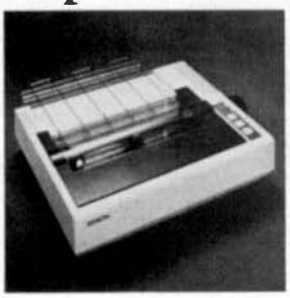
Printer		Cash	Credit Card
IDS	445	632.50	650.00
IDS	460G	852.50	876.00
IDS	560G	1094.50	1124.50
Centronics	739	519.00	533.50
Centronics	739.3	641.50	658.79
TI	820	1738.00	1785.50

All printers and terminals are shipped freight collect and there are no COD's.

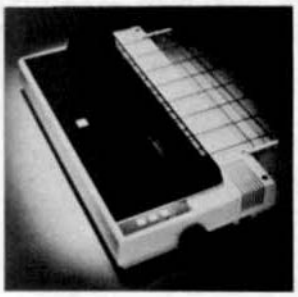
FRANK HOGG LABORATORY
130 MIDTOWN PLAZA
SYRACUSE, NY 13210 (315) 474-7856

EPSON DOT MATRIX PRINTERS

The Epson MX-80



The Epson MX-100



REDUCED PRICE SPECIALS

80-Column

~~$495.00~~ $455.00

MX-70 ~~$895.00~~ $355.00

136-Column

~~$798.00~~ $725.00

MX-80 FT ~~$626.00~~ $575.00

FACTORY FRESH — LATEST PRODUCTION — BRAND NEW

SWTPC-Motorola, MP32 Dynamic Memory Board **ONLY $199.00**

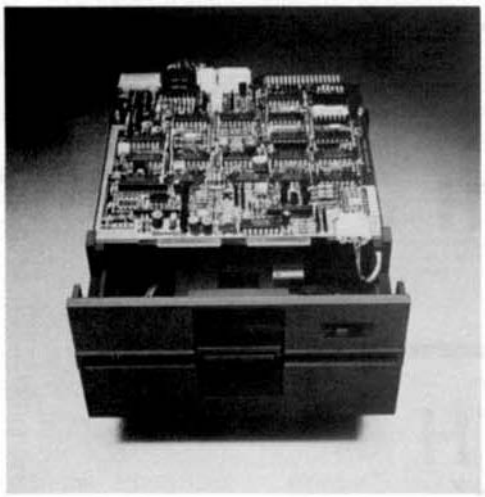
Assembled & Tested

Thousands sold at $650.00 Save over $450.00

1 MHZ - No extended addressing

Can be set up for $0-7FFF or 8000-FFFF

Limited Quantity (less than 50)



**** DISK DRIVES & CABINETS W/PS ****

5 1/4"

TEAC - Single Sided, Double Density, 40 Track.....$249.95

TEAC - Double Sided, Double Density, 40 Track.....$349.95

CABINET - Single Drive with Power Supply.....$ 79.95

CABINET - Double Drive with Power Supply.....$ 99.95

CABLE - Single Drive.....$ 24.95

CABLE - Double Drive.....$ 34.95

* NOTE - When ordering cables please specify S50 Bus or Other!!!

Call or write for disk controller Board information.

MasterCard VISA

SOUTH EAST MEDIA

P.O. Box 794 Chattanooga TN 37443

1-615-842-4601

SHIPPING: - Memory Boards or Cables, Add $4.50; Printers, Add $7.50
- Disk Drives or Cabinets with Power Supply, Add $7.50

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

ASSEMBLER TECHNIQUES

In the February column, I presented a technique for allocating variables on the stack. I didn't go much into why you might want to do that. Among the responses I received about that column, several people realized that this is a way to allocate what are called "local variables" in a subroutine. Pascal OR "C" users will be familiar with the concept, but those who have programmed in BASIC or Assembler only, may not understand their significance. Suppose you are writing a very large program. We all know that in order to keep things managable, you should break the program up into small units. Sometimes the easiest way to do that is to write subroutines to do each "functional block" of the program, and then to write a "main program" that does little more than call the various subroutines in the correct order to execute the program.

Many times, a subroutine will need what you could call a "scratchpad" memory area. Some intermediate calculation results must be stored for later use within that subroutine. The important point is that these variables are needed only until the subroutine has done its job and the answer is calculated. This allocation on the stack is the ideal way to set up local variables. It in fact, almost guarantees that the subroutine will work in a recursive program, in which it may call itself. That feature is perhaps the main reason for having local variables in Pascal or "C". Another possible reason for such allocation only during the subroutine, with deallocation at the end, is that the "scratchpad" area is released for use by another subroutine in the program, resulting in reduction in the amount of memory needed for the program. This may be particularly important in a dedicated application where the program is in PROM and the only need for RAM is for the stacks and the variables.

Now, with that in mind, I am going to present the best idea that was sent to me in response to my example. It was sent by Donald Korte of Flint MI, about 60 miles from my home. Don took the allocation of variables via the SET feature of the assembler a bit farther and defined two Macro's to do all the work. I'll stick to the nomenclature used by Don and present them as he sent them to me. The first simply initializes the variable ARGS to zero. The second assigns the offsets on the stack for each variable as it is defined by the user. The program MACTEST included here, contains the macros and shows how they are used. I have assembled the program with the macro expansion enabled so you can see how each variable is assigned its space. Since ARGS is left with the count of all the space allocated after the last call to DEFARG, it may be used to deallocate the space on the stack at the end of the subroutine. Don't miss the point that you can use these macros an unlimited number of times as you add subroutines to your program. You would probably not assemble with the Macro Expansion assembler option on, since after a couple of uses you know what the macros do.

If you assemble this source, you may find, as I did, that some early versions of the TSC 6809 Macro Assembler object to an argument being in the label field in a macro. The manual indicates that this is legal, and the macro works properly, but I always get an error message when I assemble it. Perhaps later versions of the Assembler have the problem resolved.

Don, thanks for the tip. I guess most of us have simply ignored the Macro capability of our assembler. Along that line, I will quote from a letter from Nigel Bennee of Lucidata. This is not an exact quote, but Nigel said that this (TSC Macro Assembler) is the second best assembler he has ever worked with. He added that the best one was on a large computer installation in which the ASSEMBLER (not the whole system) was leased for $1000 a month! Things have come a long way in the computer area in the past few years. Nigel also sent me some math routines for a 16 by 16 bit multiply and divide, which have rather complete checking for overflow. He related that he was shown Pascal MT on a Z-80 system, and that it accepted 40000 as an integer and treated it as though it were -7232 without giving an error message. Nigel wants to know what earthly good such software could be for any real world application.

Since the discussion of a 16 by 16 multiply would be a logical extension of our present variable allocation discussion, let's first go into how to do multiple precision multiplies with the MUL instruction, and then write a routine that is entered with two numbers on the stack and returns with the result there. We will first test this Multiply routine by using Flex routines to input and output the numbers. Later we will make it an independent subroutine that expects to be passed the arguments on the user stack.

The MUL instruction multiplies the value in A by the value in B and leaves the result in D. It is an unsigned multiply. If you want to do signed arithmetic, the easiest way is to convert negative numbers to positive and save the sign in a flag somewhere. First, lets assume the numbers are positive. Later we can add the sign logic and the complement and add subroutine to convert input numbers to sign and magnitude and to convert the result to two's complement if necessary. Unfortunately, as most things work out, it is easier to multiply and divide if the variables are in the sign and magnitude form, and it is easy to add and subtract if they are in the twos complement form. (So what else is new.) You may remember that in most integer modes of compilers and BASIC, the range of numbers is from -32768 to +32767. In twos complement form, 32767 is 01111111 11111111, and -32768 is 10000000 00000000. When two integers are multiplied, the product must not exceed these limits, or the result would give erroneous information in the sign bit. Therefore, we can have the two arguments take the extreme values 32767 and 1, or intermediate values down to the limit of the equal values 181 and 181, and still have the result not overrange.

To refresh your memory, take pencil and paper and multiply 56 by 78. You have two "partial products" which you added to get the result 4368. Now try it a different way. Multiply the two low order digits and get 48. Now take the product of 8 and 5 (high order times low order) and write 40 moved one place to the left below 48. Now take the other cross product 6 times 7 and write the 42 below the 40. Last take the product of the two high order digits 5 times 7 and write the 35 moved one place to the left of the previous products. Now add 48 420 400 and 3500 and get the same result 4368. That is exactly the way the 16 bit multiply works. Low times Low, add the two "cross products" high times low and low times high shifted 8 bits left, and add the high times high shifted 8 more bits left. Remember that carry may be generated anywhere in the process of adding the partial products, and be certain to handle these.

The program MULTI does this and uses the FLEX routines INDEC and OUTDEC to allow you to input decimal numbers and get the result in decimal. The consequence of using those routines to test the multiply, is that you have to input the two numbers on the command line: MULTI.BIN,12,12 will get you 144 if the multiply is correct. Note that this is only good for unsigned numbers, but INDEC and OUTDEC only work for unsigned integers anyway, so they are useful for the test of the "raw multiply" routine. We've used the Macros to define our bytes for the two 16 bit arguments and the 32 bit result. Of

course result is only valid if the top 16 bits are all zero, and we check for that and print an error message if overflow occurs. If this routine were to be incorporated in the middle of a math package, for example, you would probably have the main program set up the user stack with space for the result, and the two arguments on the stack. You would do the calculation and return, and the main program would do the clean up of the U stack pointer after recovering the result of the multiplication at 6,U. I hope this is sufficiently clear to be of use to some of you. I promise that we will develop MULTI further next month, adding the steps necessary to handle signed integers.

MORE ON PASCAL

One reader wrote me and wanted to know how to get around the lack of the WITH statement in Lucidata Pascal. WITH is only a convenience in accessing the fields of a RECORD variable in Pascal, and it is not necessary to use it, though it may be a great convenience. Since RECORDS are standard in Pascal and any good Pascal text covers them, I will assume that you know how to declare a record type. Suppose you have a record LABEL with three fields declared as arrays of character, NAME, ADDRESS, and CITY. You can access these parts by using WITH as in the following sample of code.

```
WITH LABEL DO
BEGIN
  NAME := 'ALEXANDER G. BELL   ';
  ADDRESS := '1234 5TH ST.     ';
  CITY := 'LITTLE SPRINGS    ';
END;
```

Without the WITH facility, you simply use the whole name as below:

```
LABEL.NAME := 'ALEXANDER G. BELL   ';
LABEL.ADDRESS := '1234 5TH ST.     ';
LABEL.CITY := 'LITTLE SPRINGS    ';
```

For records with few fields, it may take less typing to use the second method even if WITH is available. Lucidata points out that you can use a short name for the record. For example in this case you could simply use L for the record name, and the access would be L.NAME, L.ADDRESS, and L.CITY.

SETS

The subject of SET theory is rather near at hand to me at this time. My daughter is now learning Algebra in 9th grade. I learned Algebra in 9th grade about 30 years ago, and I have been using it ever since, but I have great difficulties with helping my daughter because of the great preoccupation with SETS that has developed in the teaching of mathematics since the time I learned it. I frankly don't see the value of tieing set theory so closely with Algebra.

Anyway, a SET is a defined group or collection of items. For example the set of positive integers from 1 to 5 (inclusive) is [1,2,3,4,5]. That is simple enough, isn't it. The set of EVEN positive integers in that range is [2,4] and the set of ODD integers is [1,3,5]. Here, the Symbolic logicians and the Programming Language writers differ because of the ASCII character set. Lets call our two sets ODD and EVEN.

Set theory defines a couple of functions they call the UNION and INTERSECTION of sets. The UNION is what we programmers would call the OR function. The UNION of the two sets ODD and EVEN as defined above, is [1,2,3,4,5]. In Pascal we use the + sign to represent UNION. BOOLEAN algebra uses + (addition) to represent the OR function. The UNION of two sets is the list of all the items that are in either of the sets or both of them. That is, the UNION is a plain OR function, not an EXCLUSIVE OR. Anyone out there need to have explained what the INTERSECTION is? Of course, it is the AND function of the two sets. The INTERSECTION of ODD and EVEN is [], the NULL set. BOOLEAN Algebra uses multiplication to represent the intersection. In Pascal, of course, the multiplication symbol is the asterisk or star (*). How do we define a set in Pascal? First we must have an ennumerated type.

```
TYPE
  COLOR = (RED,BLUE,GREEN);

VAR
  COLORS : COLOR;
  MIX  : SET OF COLOR;
```

What is the difference between COLORS and MIX you might ask. COLORS may take on the value RED or the value BLUE or the value GREEN. It has three possible values. MIX, on the other hand may take on the values shown below:

```
RED
BLUE
GREEN
RED BLUE
RED GREEN
BLUE GREEN
RED BLUE GREEN
[]
```

While COLORS may have three values, MIX may have 8. In general, if the enumerated type contains N elements, the set of that type may have 2^N values.
Pascal lets us assign values to sets as MIX := [RED, GREEN];. While space doesn't permit much in the way of a complex example here, imagine that you have a collection of articles from such magazines as 68 Micro Journal, and you want to index them for future reference.

```
TYPE
  ATTRIBUTE = (6800,6809,ASEM,COMPILE,HARDWARE);

VAR
  ATTRIBUTES : SET OF ATTRIBUTE;
  SEARCH   : SET OF ATTRIBUTE;
```

Now suppose you set up an array of attributes, and associate each article with an array index. You can then input the values for each array entry, which will categorize the article associated with it. Later, if you want to find all the articles that refer to 6809 high level compilers, you can search the array:

```
SEARCH := [6809, COMPILE];
 FOR INDEX = 1 TO MAX DO
  IF ATTRIBUTES [INDEX] * SEARCH = SEARCH
   THEN WRITELN (INDEX);
```

When this is run, you will have a list of the index number of all articles that have the attributes 6809 and COMPILE. By now you can think of other uses for SETS.

MY FACE IS RED DEPARTMENT

I received a call recently from Jesse Saib in New York, to point out that I had made an error. In my March column, I presented an example program called DUMPROM. Line 90 of that listing is LEAY -1,Y which decrements a counter. The next line contains code that works, but is unnecessary. Line 91 reads CMPY #0 and has the comment: LEAY DOESN'T SET ZERO FLAG. That is an error. The X and Y registers do set the zero flag when their contents become zero. They don't have any effect on any other flags. Incrementing or decrementing U or S has no effect on any flag. My program would obviously run just as well without line 91. Thanks, Jesse for pointing out my error. When in doubt read the instructions. That folding green instruction set summary from Motorola has it all if we will take the time to look. Any other sharp eyed readers who catch a dumb one like that are welcome

to tell me about it, and I will correct the error in a later column.

TSC MOVES

Technical Systems Consultants, who were the prime mover in getting the 68xx off the ground by supplying good software, have moved from their old address in West Lafayette, IN. to new facilities in North Carolina. In case you have not yet gotten their new address it is:

Technical Systems Consultants, Inc.
111 Providence Road
Chapel Hill, NC 27514

I guess when virtually all of your business is "mail order" you don't have to endure the Midwestern winters! Congratulations all you folks at TSC on your relocation and new facilities. We "northerners" will envy you come next winter.

I recently communicated with Dan Vanada of TSC, and he had encouraging words for Flex users. While most if not all of TSC's new software will first be developed for Uniflex, Dan assures me that FLEX09 versions will follow. Among the projects listed are a Fortran compiler, a "C" compiler, and a screen editor. Dan also indicated that their new Relocatable Assembler and Linking Loader that are now out in the Uniflex version, will be available soon in the Flex09 version. News for the 6800 users is not quite so good. Dan assures me that they will continue to support FLEX2 and the software already available for it, but that no new 6800 software will be developed.

I guess you could call that the pains of progress. You 6800 users have three choices. First you might wait and see if some other suppliers continue to develop 6800 versions of their software (Star Kits and Frank Hogg Laboratories are good examples of suppliers who are doing that). Second, you could decide to switch to 6809 at this point, which means rather costly purchasing of new software. Third you might decide to hold out a bit longer and wait for the next major leap in technology, the 68000. If you are not a "collector" (of software) and can have fun with what you have for your 6800 system, perhaps the latter is the best approach for now. With memory prices going down so fast, those 256K systems might just get reasonably priced and soon.

```
 1.00= NAM MULTI
 2.00= TTL MULTIPLY PROGRAM
 3.00= OPT PAG
 4.00= PAG
 5.00=*
 6.00=*
 7.00=WARMS EQU $CD03
 8.00=INDEC EQU $CD48
 9.00=OUTDEC EQU $CD39
10.00=PSTRNG EQU $CD1E
11.00=PCRLF EQU $CD24
12.00=*
13.00=INIARG MACRO
14.00=ARGS SET 0
15.00= ENDM
16.00=*
17.00=DEFARG MACRO
18.00=\1 SET ARGS
19.00=ARGS SET ARGS+\2
20.00= ENDM
21.00=*
22.00=* LOAD ADDRESS 0
23.00=*
24.00=START INIARG
25.00= DEFARG HI,* HIGH ORDER BYTE OF ARG 1
26.00= DEFARG LI,* LOW ORDER BYTE OF ARG 1
27.00= DEFARG H2,* HIGH ORDER BYTE OF ARG 2
28.00= DEFARG L2,* LOW ORDER BYTE OF ARG 2
29.00= DEFARG MSB,* MOST SIGNIFICANT BYTE OF RESULT
30.00= DEFARG NMSB,* NEXT MOST SIGNIFICANT BYTE OF RESULT
31.00= DEFARG NLSB,* NEXT TO THE LEAST SIGNIFICANT BYTE OF RESULT
32.00= DEFARG LSB,* LEAST SIGNIFICANT BYTE OF RESULT
33.00=*
34.00=* MAKE ROOM ON THE STACK FOR THE VARIABLES
35.00=* THEN POINT U AT THE VARIABLES SO THAT THEY MAY BE
36.00=* REFERENCED WITHIN ANY SUBROUTINE IN THIS SECTION OF CODE.
37.00=*
38.00= LEAS -ARGS,S
39.00= TFR S,U
40.00= JSR INDEC GETS ARG 1 IN X REGISTER FROM COMMAND LINE
41.00= STX HI,U
42.00= JSR INDEC GETS ARG 2 IN X
43.00= STX H2,U
44.00= LDD #0
45.00= STD MSB,U CLEAR RESULT AREA
46.00= STD NLSB,U
47.00=*
48.00=* NOW MULTIPLY
49.00=*
50.00= LDA L1,U
51.00= LDB L2,U
52.00= MUL LOW ORDER TIMES LOW ORDER
53.00= STD NLSB,U NEXT TO THE LEAST SIGNIFIANT BYTE
54.00= LDD L1,U A GETS L1 AND B GETS H2
55.00= MUL GET A HIGH AND LOW ORDER 'CROSS PRODUCT'
56.00= ADDD NMSB,U NEXT TO MOST SIGNIFICANT BYTE
57.00= STD NMSB,U
58.00= BCC MULT1
59.00= INC MSB,U CARRY IF NECESSARY
60.00=MULT1 LDA HI,U
61.00= LDB L2,U
62.00= MUL OTHER HIGH AND LOW ORDER 'CROSS PRODUCT'
63.00= ADDD NMSB,U SAME POSITION AS LAST 'CROSS PRODUCT'
64.00= STD NMSB,U
65.00= BCC MULT2
66.00= INC MSB,U  HANDLE CARRY IF NECESSARY
67.00=MULT2 LDA HI,U
68.00= LDB H2,U
69.00= MUL HIGH ORDER TIMES HIGH ORDER PRODUCT
70.00= ADDD MSB,U
71.00= STD MSB,U
72.00=*
73.00=* NOTE THAT IF BOTH ARGUMENTS WERE WITHIN RANGE SUCH THAT
74.00=* OVERRANGE COULD NOT OCCUR. THE RESULT OF THIS LAST MULTIPLICATION
75.00=* WOULD HAVE TO BE ZERO.
76.00=* NOW WE CAN TEST FOR OVERFLOW.
77.00=*
78.00= CMPD #0
79.00= BED OUTNUM
80.00= LEAX ERROR,PCR POINT AT ERROR MESSAGE
81.00= JSR PSTRNG PRINT IT
82.00= BRA EXIT RETURN TO FLEX
83.00=OUTNUM JSR PCRLF
84.00= LEAX NLSB,U POINT AT RESULT FOR OUTDEC
85.00= JSR OUTDEC
86.00=EXIT LEAS ARGS,S
87.00= JMP WARMS  NORMALLY WOULD HAVE RTS HERE
88.00=*
89.00=ERROR FCC / OVERFLOW - ARGUMENTS TOO LARGE./
90.00=      FCB 4
91.00=*
92.00= END START
```

```
 1.00= NAM MACTEST
 2.00= TTL TEST OF VAR ALLOC,MACROS
 3.00=*
 4.00=* THIS IS A TEST OF TWO MACROS TO ALLOW EASY ALLOCATION
 5.00=* OF STACK SPACE FOR LOCAL VARIABLES
 6.00=*
 7.00=WARMS EQU $CD03
 8.00=GETHEX EQU $CD42
 9.00=*
10.00=INIARG MACRO
```

```
11.00=ARGS SET 0
12.00= ENDM
13.00=*
14.00=DEFARG MACRO
15.00=&1 SET ARGS
16.00=ARGS SET ARGS+&2
17.00= ENDM
18.00=*
19.00= ORG $C100
20.00=*
21.00=START INZARG
22.00= DEFARG FIRST,2
23.00= DEFARG LAST,2
24.00=*
25.00=* MAKE ROOM ON THE STACK FOR THE VARIABLES
26.00=* THEN POINT U AT THE VARIABLES SO THAT THEY MAY BE
27.00=* REFERENCED WITHIN ANY SUBROUTINE IN THIS SECTION OF CODE.
28.00=* THE FOLLOWING IS THE FILL MEMORY CODE AS AN EXAMPLE
29.00=*
30.00= LEAS -ARGS,S
31.00= TFR S,U
32.00= JSR GETHEX
33.00= STX FIRST,U
34.00= JSR GETHEX
35.00= STX LAST,U
36.00= JSR GETHEX BYTE TO FILL WITH
37.00= TFR X,D NEAT TRICK FROM FRANCIS VAN HORN
38.00=*
39.00=* DO FILL PROGRAM AS DESCRIBED IN PREVIOUS COLUMN
40.00=* FILL BYTE NOW IN B REGISTER
41.00=*
42.00= LDX FIRST,U
43.00=LOOP STB ,X+
44.00= CMPX LAST,U
45.00= BLS LOOP
46.00=*
47.00=* NOTE THAT THE 'S' MUST BE CLEANED UP TO PREVENT 'CREEPING STACK'.
48.00=*
49.00=* IF A PROGRAM USES 'U' CONSISTENTLY AS SHOWN HERE, IT IS INITIALIZED
50.00=* BY EACH SECTION OF CODE, AND NEED NOT BE RESTORED AT THE END OF
51.00=* ANY SECTION.
52.00=*
53.00= LEAS ARGS,S
54.00= JMP WARMS
55.00=ZZSIZE EQU *-START AUTO CALCULATION OF BYTES OF CODE IN HEX
56.00= END START
```

OS-9 NOTES

From Ray Cadmus
600 W. Lee
Moberly, Mo. 65270
(816)263-1228 - Home
(816)263-6693 - Office

Today I'll give you a peek into my dis-organized mind and just ramble a bit. I haven't gotten enough feedback as yet to get a feel for the experience level of the average reader. (Hint, hint!) So, to be on the safe side, I'll begin with some of the basics - some of the dumb things I had to learn the hard way. We'll look at setting up a basic system disk and present a couple utilities that make that chore a little easier. For what it's worth, I'll present my approach to developing utility programs and using existing ones. Then, for those people who insist on writing big programs for little machines, we'll look at a simple method of using overlays in BASIC09.

Just a little aside here - You will find one central theme running through the way I do things. I think it's called basic laziness, or as I saw it put so well recently "creative sloth". I'm firmly convinced that lazy people make the best programmers. We work much harder at finding easy ways to do things and really, isn't that what computers are for?

SETTING UP A SYSTEM DISK - I received my distribution programs on single sided, single density 35 track disks. The first thing to do is copy EVERYTHING. No problem, BACKUP is quick and easy. Much faster, by the way, if you give it lots of room to work, ie. BACKUP #30k or so. Then I formatted a double sided work disk and ran BACKUP again to get a working copy. "TILT!" BACKUP works only with identically formatted disks. At this point I did the only thing I could, I wore blisters on my fingers going COPY COPY COPY file by file to get every thing on the new disk. Then I discovered that the serial printer device descriptor used a different address than I wanted for my printer, so I spent the first couple weeks using DEBUG to change the pointer every time I booted the system.

Next - to add insult to injury - I received a new release of the OS9 system so I went through the same thing again with all the system files plus all the new stuff I had added by now. At about this same point in time I discovered COBBLER and SYSGEN and realized that I could fix my printer address problem and do some other neat things like speed up the disk stepping times. This was great but, because the boot module must be contiguous as the first file on disk I faced the COPY COPY COPY bit again. Enough! I wrote DCOPY to copy a directory's worth of files at a time. At that time the +DIR option was not available so the first version was a little messy, so this is the second version. Eventually I'll rewrite it in assembler but for now it works. The companion program PURGE is the same thing in reverse. It allows easy cleanup of a directory. I believe the programs are pretty much self-explanatory so I won't bore you with a description.

Here is something to keep in mind when making heavy use of the utilities. LOAD all the utilities you will need first. This keeps the system from loading them each time you call them. LOAD LOAD first, then load COPY, load DIR etc. I usually execute a shell file that looks like:

```
TX
LOAD LOAD
LOAD DIR
LOAD COPY
LOAD LIST
etc...
```

You'll note that this capability sets OS9 apart from most other operating systems. On most systems the system disk must be in the primary drive at all times. With OS9 you can do a series of loads, then throw the disk away and still run just fine.

I write the occasional utility program when the need comes up. I also write some just for fun. Application programs I write for a living, utilities for me. Here are a few guidelines which (unfortunately) I don't always follow. Make the program self explanitory and fully prompting. I often find that, when I violate this, I have to list the program source to find out how to run it. There have even been times that when I list the source I discover that I neglected to comment properly and still can't figure out what I did or it does. It sure hurts to have to throw one away and start over because I didn't take the time to do it right the first time. Next - write it in as high a level language as will do the job adequately. It may not be super efficient, but it's lots easier to get going and to refine into a smooth package. Then, if it will get lots of use, you may want to rewrite it in assembler to speed things up. Right now I start out in BASIC09 and many never get beyond that point. When I get a C compiler I'll probably use that for most. I do have a PASCAL compiler, but BASIC09 does so many things so easily that I rarely use it. Besides, after using BASIC09 and C, PASCAL is a PAIN!

Now - Big BASIC09 programs on little machines. Have you looked at the size of BASIC09? That thing is BIG! If I recall correctly it takes about 27k leaving

about 17k as program and data space on my 56k machine. Now keep in mind that I don't have the smaller run time module yet, so this may seem dumb to those of you that do. When I started running out of room on a major project, I rewrote the program breaking it up by functions. Remember that one procedure can call any other and pass paramaters (pointers) to shared data. As long as a procedure has been "packed" into the cmds directory a run statement will find and load it as necessary. The trick is to "kill" it when you are finished with it for a time. Just remember to define the shared data in the permanently resident part of the program so everyone can get a crack at it.

Last month I warned about Z19 terminals giving problems with spurious data and system hangs. I've since heard a couple reports of Televideo terminals giving the same troubles. In investigating the problem further, it seems that there is nothing inherently wrong with the terminals - just that some types are more sensitive than others to power line noise. I've been able to use a Z19 ok by filtering the AC lines HEAVILY. A normal line filter will not correct the problem, it took a line filter in conjunction with a constant voltage transformer to do the trick. The Gimix mainframe I use did not require filtering - just the attached peripherals.

NEW PRODUCTS - I just received some data on a series of file transfer utilities by META LABS that look interesting. They are designed to allow reading and writing FLEX files under OS9. I hope to have a detailed report for you next month.

Our mail service is a wonderful thing. I started receiving phone calls and letters on this column a week before I saw the first one in print. Please - keep the calls and letters coming. I've been writing blind for three months now and I'm beginning to reach the point where I NEED your suggestions and submissions to keep this thing going. I hope that we can make this an OS9 forum, not just my personal soapbox.

Bye

```
PROCEDURE DCOPYS
 0000      REM COPY DIR FILES
 0011      REM SPACE TO COPY - ANY OTHER KEY TO BYPASS
 003B
 003C      DIM C$:STRING[1]
 0048      DIM P:BYTE
 004F      TYPE REC=NAME:STRING[29]; MISC:STRING[3]
 006A      DIM DREC:REC
 0073      INPUT "FROM DIR > ",FD$
 0086      INPUT "TO DIR > ",TD$
 0097      CHD TD$
 009C      PRINT "SPACE TO COPY - ANY OTHER KEY TO BYPASS"
 00C7      PRINT
 00C9
 00CA      ON ERROR GOTO 10
 00D6
 00D1      OPEN #P,FD$:READ+DIR
 00DD      WHILE NOT(EOF(#P)) DO
 00EB 5      REM READ NEXT DIR ENTRY
 0101        GET #P,DREC
 010B        IF LEFT$(DREC.NAME,1)>" " THEN
 011E          IF LEFT$(DREC.NAME,1)<>"." THEN
 0131            RUN FIXS(DREC.NAME)
 013E            CMD$="COPY "+FD$+"/"+DREC.NAME+" "+DREC.NAME
 0164            PRINT CMD$; " ";
 016E            GET #0,C$
 0177            PRINT
 0179            IF C$=" " THEN
 0186              SHELL CMD$
 0188            ENDIF
 018D          ENDIF
 018F        ENDIF
 0191      ENDWHILE
 0195      CLOSE #P
 0198      END
 019D 10   REM ERR ROUTINE - BYPASS & TRY AGAIN
 01C3      PRINT "ERROR = "; ERR
 01D1      GOTO 5


PROCEDURE PURGE
 0000      REM !!!!!!!! CLEAR SELECTED FILES FROM DISK !!!!!!!!!
 0034
 0035      DIM ACTION:STRING[1]
 0041      DIM D:BYTE
 004B      TYPE REC=NAME:STRING[29]; MISC:STRING[3]
 0063      DIM DREC:REC
 006C      RUN CLS
 0070      PRINT  \ PRINT
 0074      INPUT "DIRECTORY TO PURGE ",D$
 008F      CHD D$
 0094      PRINT
 0096      PRINT "ENTER <Y> TO DELETE - ANY OTHER KEY TO BYPASS"
 00C7      PRINT
 00C9
 00CA 500  REM !! MAIN !!
 00DA      ON ERROR GOTO 510
 00E0      OPEN #D,D$:READ+DIR
 00EC      WHILE NOT(EOF(#D)) DO
 00F7 505    GET #D,DREC
 0104        RUN FIXS(DREC.NAME)
 0111        IF ASC(LEFT$(DREC.NAME,1))<>0 THEN
 0124          IF LEFT$(DREC.NAME,1)<>"." THEN
 0137            PRINT DREC.NAME; " ";
 0144            GET #0,ACTION
 014D            IF ACTION="Y" THEN
 015A              DELETE DREC.NAME
 0162              PRINT " OK"
 0169            ELSE
 016D              PRINT
 016F            ENDIF
 0171          ENDIF
 0173        ENDIF
 0175      ENDWHILE
 0179      END
 017B 510  PRINT "ERROR = ",ERR
 018C      GOTO 505
```

"C" User Notes

Norm Como
3 Pryor Road
Natick, MA 01760

The process of getting from a source file written in C to the final executable binary can follow a number of paths. One of the cleanest ways is when the output of the compiler is the input to a relocating assembler that can

be used with a linker, especially when the compiler chains to the assembler automatically. Ideally, you start the compiler which takes the source code and produces the assembler source. It then chains to the assembler which produces a relocatable object module. At this point the process stops. You can then link that object module with the library and any other modules that may be needed.
Unfortunately for FLEX and OS-9 users no standard relocating assembler exists. I have seen the ad for the CINCITEK relocating assembler, and I know that TSC has one for UNIFLEX which should be migrating to FLEX fairly soon. But they weren't there when the various software vendors started coming out with 6809 C compilers.
This means that all the presently available compilers must contend with absolute assemblers. Usually the assembler of choice is TSC's for FLEX and Microware's for OS-9. What makes this all so nasty is what it forces you to do in order to link separately compiled sources into an executable binary.
How do we do it? As always, the answer depends on whose compiler you own. If you own the Introl compiler life approaches being a piece of cake. If you happen to own the Word's Worth compiler things get a little harder. Finally, if you own either Intersoft's or Dugger's product things get harder still. So let's start from easy and work to hard.

BUILDING INTROL PROGRAMS

The Introl compiler comes with its own absolute assembler, linking loader and librarian. What's that? A linking loader with an absolute assembler? Yes, that's correct. Introl did a very clever thing. The compiler forces the assembler to take a relocatable binary module. It does this by a combination of position independant code and the rather adroit use of FCB's, FDB's, FCC's and module labels to produce the relocation data tables. The only real compromise is the handling of calls to functions that exist outside of the module. A jump table is built at the end of the module. During the link, these tables are filled in with the real addresses of the functions.

The command sequence to get from the C source to the executable would be

```
+++EDIT SOURCEFILE
+++PP SOURCEFILE
+++MLINK SOURCEFILE
```

The PP (preprocessor) command starts the compile, which takes a total of five passes to produce the relocatable object module. The MLINK command links the module to the standard library, which need not be named if it resides on the working disk. It pulls in only those functions that are actually called in the user program. What could be easier? You now have SOURCEFILE.CMD on your working drive. Since Introl offers their compiler on 8 inch drives only and I use single density 40 track minifloppies, I have another step to do.
The review copy that I received came on three 5 inch diskettes that were copied by someone at 68 Micro. The four programs needed to compile a module would not all fit on one single density minifloppy. So I merely made my own assembler that was nothing more than a jump to FLEX's warm start. On one disk I put PP.CMD, P1.CMD, P2.CMD, my own assembler and a few utilities. On another disk I put the real assembler, the linker, the librarian, the library and a few utilities. One more was needed for the editor. So an evening's session at C requires three system disk. My command sequence is the following

```
Insert "edit" disk
   +++EDIT SOURCEFILE
Insert #1 compiler disk
   +++PP SOURCEFILE
Insert #2 compiler disk
   +++A09 SOURCEFILE
   +++MLINK SOURCEFILE 0.STDLIB.LIB
```

Since the output of the code generator, P2.CMD, is just a file called SOURCEFILE.A09, the process can be stopped at that point and the disk changed. This system works just fine. Notice that I must specify the standard library file since I put it on my system disk. This saves room on the working disk for the source file and the intermediate files produced by the preprocessor and the parse tree generator, which are PP.CMD and P1.CMD respectively.
Normally I wouldn't belabor an exception like this but some rumblings indicate that Introl may someday offer their compiler on minifloppies.

The standard library is quite complete and comes with three different levels of runtime environment. One that sets up the cpu state, another that also parses the command line into arguments and one that adds the ability to redirect standard I/O to and from files. The latter is the default. You make your choice when you link the program. Since the library is completely binary, it takes up less than forty blocks. That's not very much when you compare it to the space needed if the library is kept in assembler source code. There is a librarian that lets you add your own function or new version of existing functions.
All in all, Introl are to be congratulated for a very fine product that incorporates some clever programming.

BUILDING WORD'S WORTH PROGRAMS

Word's Worth did something similar to Introl. Their compiler's output code includes FCB's, etc to allow the assembled module to be linked with RLOAD. RLOAD is a pseudo-linker that was published in previous issues of this magazine(1). Where RLOAD falters is that all the functions in a library file are added to the program, even if only one is actually used. Judicious granulation of library functions over separate files can overcome this to a great degree. Another shortcoming, relative to a real linker, is that there is no library per se if you are trying to minimize the size of a program by following the previous advice. In this case all your compiled and assembled functions exist as binaries, but in unique files. Moving a library from disk to disk, and building a link file is made less convenient.
Now let's set up some assumptions about our "Word's Worth environment". We have separated our standard functions into four files, called

```
C1.IB.LIB       -- the minimum library, used
                   used with every program. It
                   contains the runtime package
                   and terminal i/o functions
                   getchar(), printf(), etc.

CHAROPS.LIB     -- character functions such as
                   isalpha(), isdigit() etc.

FILEIO.LIB      -- the file operation functions
                   such as fopen(), getc(), etc.

STRING1.LIB     -- the primary string functions
                   like strcpy(), strcmp(), etc.
```

The command sequence to get from the source file to the executable binary would look like

```
+++CC      (which prompts for files)
+++COPT    (prompts for filenames)
+++ASMB ASMBFILE,BINARYFILE,+YP10
+++RLOAD LINKFILE.LNK
```

The compiler will prompt you for the filenames and some other things. Word's Worth recommends the following file extension defaults

```
SOURCEFILE.C  source code to the compiler
ASMBFILE.ASM  assembler code from the compiler
ASMBFILE.OPT  assembler code from the optimizer
```

```
BINFILE.REL   binary from the assembler
LINKFILE.LNK  the RLOAD directive file
```

The RLOAD directive file contains the name of the .CMD file to be built and a list of all the object files that you want it to load in. While this is not quite as convenient to use as Introl, it's still fairly flexible. And again, the "library" is kept as binary code, which save greatly on disk space. The standard functions supplied with the package include most of those mentioned in C NOTES 4.
The COPT command is optional. It is a "peephole" optimizer that opens a window of a few line into the assembler code file produced by the compiler. It looks for certain sequences of instructions, and replaces them with more efficient sequences. Word's Worth claims that the savings will be between 15 and 35 percent. The optimizer is itself written in C. Both the binary and the C source are supplied with the package.

BUILDING DUGGER AND INTERSOFT PROGRAMS

Intersoft and Dugger both require the most "work" to get from a C source to an executable program. Intersoft takes more work initially since all their library functions come as C sources. So you will have to compile them separately. The user interface of both these compilers is very much like that of the original Small C. The compiler is invoked without any file name. It prompts you for the name of the output file and then prompts for the name of the input file. After that file has been compiled you are prompted for another input file. If you enter a filename, then that file is compiled also. The process is stopped when you enter just a <cr> for a filename. A minor note is that Intersoft prompts for the name an input file first, then the output file, and finally more input files.

In this manner separate C sources may be linked together by the compiler. There is one slight drawback. If you #define the same symbol in more than one input file you will get errors on the Intersoft compiler. This I found out when I tried to take the C sources for their standard functions and build then into the four library files that I mentioned earlier.
Let us again assume that you have built up the four library files that contain the standard functions. The simplest way build programs is to submit a "build" file to the assembler which contains a series of LIB's to bring in all the necessary files. Take a text editor for example. It may consist of three modules; a module that updates the text buffer, a module that updates the screen and a module that interprets and dispatches user commands. The build file could be called EDITOR.BLD and would probably contain the lines

```
ORG $0100
LIB TEXT.TXT
LIB SCREEN.TXT
LIB DISPTCH.TXT
LIB CHAROPS.LIB
LIB FILEIO.LIB
LIB STRNGI.LIB
LIB CLIB.LIB
END CINT
```

To build the file all you have to do is type in the command

```
+++ASMB EDITOR.BLD
```

In effect, you have made the assembler your linker. This is not the most efficient way of doing things since the library exists as source code and that means a lot of disk space. On the other hand, at least this method let's you break a program into modules. If there are any bugs to be fixed, or updates, all you have to do is recompile the affected module and reassemble the program. Incidently, this technique is very nicely illustrated in the Intersoft manual. Both of these compiler come with a reasonably complete set of functions.

MY OWN CHOICE

I'm sure that some of you are wondering which compiler I use. Well my preference runs to Introl, Word's Worth, Dugger (release 1) and Intersoft in that order. Introl comes first because it only lacks four features of being full blown C and I am spoiled from using the BELL and Whitesmith compilers. It is the only one I am using at the moment. Word's Worth has done some nice things for a "Small C" derivative. Dugger is small and compiles a program fairly quickly, but lacks a lot of features. Intersoft is mentioned last because I just can't warm up to it. It's certainly a very usable and well constructed compiler, I just don't happen to enjoy using it.

A BENCHMARK

Many of you might also be wondering how these four compilers stack up against each other in terms of program size and execution speed. There was a benchmark of higher level languages written by Jim Gilbreath in the September, 1981 issue of BYTE magazine. As you might imagine, most of the languages tested ran on the Z80. The program was the Eratosthenes Sieve Prime Number algorithm, which was coded in a variety of languages, including C. I took the C program and modified it slightly (see the listing elsewhere) so that it would be compatable with Dugger release 1. The modification was nothing more than the replacement of FOR constructs with their equivalent WHILE construct. I deemed this fair since that's what the compiler does anyway.
What follows is a table of the results, listed in "winning" order.

	Int	WW	Dug	Isft
program run time	22	57	58	97
program size	1605	1802	1361	5573
compiler load time	83	27	13	37
compiler run time	45	9	17	22
module size	222	381	385	498
assembler load time	24	15	15	15
assembler run time	33	24	44	133
linker load time	11	6	--	--
link time	23	11	--	--

Some qualifications are needed to lend some objectivity to the test.

1) The test system consisted of a 1Mhz 6809 with 56Kb of ram and single density minifloppies.

2) All the compilers were submitted the same source file.

3) All times are rounded up to the next highest whole second.

4) All sizes are in bytes.

5) "load" time for the compiler refers to the time it took to load the compiler from the disk. It represents a constant overhead for every compile. Ditto for the assembly and link load times.

6) "run" time for the compiler refers to how long it took to compile the user program. Ditto for the assembler and linker.

I must also say that Intersoft payed a high penalty in terms of over all program size. I have done no optimizing of their run time package, which includes such niceties as command line parsing and I/O redirection. The execution time is representative though, because those features didn't enter into it.
An interesting side comment. I feel it is fair to compare a 2Mhz 6809 against a 4Mhz Z80 as the memory cycle time is 250 nanoseconds in each case. Since my system is a 1Mhz 6809, the program run time for the Introl version scales down to 11 seconds. The best Z80 (4Mhz) was 14 seconds, which occured for a variant of PL/1 called PL/1-80. The best time for a C compiler was 15.6 seconds for the Whitesmith compiler. This probably proves nothing, but at least 6809 users can go to bed with a smile on their face!

PEN-IN-MOUTH

As near as I can tell, I get off pretty easy this month. The only thing I found when I reread a copy of C NOTES 4 was that I left off the footnote. I had constantly used the the term K&R to refer to "The C Programming Language" by Ritchie and Kernhigan, published by Prentice Hall. After a while you get tired of writing out the whole title.

WHAT'S NEW

I have just received a copy of Dugger's compiler, version 2. It has added a lot of features and now supports longs (24 bits) and floats; the for loop, do-while, complex assignment operators, and logical operators. Constants may now be represented in hex or octal as well as decimal. He also breaks down strings into a series of FCB's and converts the escape characters for you. This makes writing your own printf() a lot simpler. I have not had a chance to try the compiler yet but Ron Anderson has. He has sent me copies of letters to Dugger detailing a series of bugs involving floating point operations. I am very happy to report that the documentation for version 2 is an order of magnitude better than it was for the early shipments of version 1.
Well that's it for this month. I haven't fully formed next month's contents in my mind yet. I do know that I want to start getting some more sample programs and code segments out to you. I also want to get into some of the complex data types that C allows. As well C functions to handle data structures like trees, linked lists, fifo's etc. Recursion needs some more explanation too. Looking further into the future I can see the need for covering topics like interrupt driven or pram based systems programmed in C.

NOTES

(1) RLOAD can be found in the September through November, 1981 issues of 68 Micro Journal.

ERATOSTHENES SIEVE PRIME NUMBER PROGRAM LISTING

```
/* Erathosthenes Sieve Prime Number Program in C
 *
 * As excerpted from the Sept, 1981 issue of BYTE
 * magazine.
 *
 * Modified for compatibility with Dugger "C",
 * release 1, which does not have FOR loops.
 */

#define TRUE 1
#define FALSE 0
#define SIZE 8190
#define SIZEP1 8191

char flags[SIZEP1];

main()
    {
    int i, prime, k, count, iter;

    puts ("10 iterations\n");
    iter = 1;
    while (iter <= 10)
        {
        count = 0;
        i = 0;
        while (i <= SIZE)
            {
            flags[i] = TRUE;
            i++;
            }
        i = 0;
        while (i <= SIZE)
            {
            if (flags[i])
                {
                prime = i + i + 3;
                k = i + prime;
                while (k <= SIZE)
                    {
                    flags[k] = FALSE;
                    k = k + prime;
                    }
                count++;
                }
            i++;
            }
        iter++;
        }
    outdec (count);
    puts (" primes\n");
}
```

COLOR User Notes

Robert L. Ney
5900 Cassandra Smith Rd.
Hixson, Tn. 37343

We'll make this a little shorter than normal this month because we have an article by Steve Odneal giving some of the User Accessible Routines in the DATA-COMP FLEX Conversion for the Color Computer, and an article from Tony DI Stefano on getting the Color Computer to run at Double Speed. We have a couple of new products to look at, both of them in the EXCELLENT Catagory. We'll take a look at STAR-KITS SPELL 'N FIX Dictionary for the Color Computer, and how it works (which will give you a general idea of how Dictionarys are used with a Computer), and take a QUICK LOOK at ATOMTRONICS DISK CONTROLLER BOARD which provides a Real Time Clock, a Parallel Printer Output, and an 80x24 Display Monitor output also.

SPELL 'N FIX
A TRS-80C Color Computer "DICTIONARY" Program

STAR-KITS
PO Box 209
Mt. Kisco, N.Y. 10549

Disk or Cassette - $69.29

Requires 32K RAM

We mentioned Pete Starks' SPELL 'N FIX Dictionary Program last month; we received the Color Computer Disk version of it a few days ago. This Program is written to run with the normal Color Computer Operating System, and is the first "Dictionary" Program for the machine that we know of. STAR-KITS has had this Program available for the FLEX Operating System on 6800 and 6809 machines for some time now, so this is not another of the "get something out for the Color Computer, quick" programs. We have been using the FLEX Version for 6 or 8 months, and have watched it progress into the excellent product that it is. The Program is "self-prompting", makes use of a "compressed" Dictionary to conserve Disk space, allows the addition and/or deletion of Dictionary Words, etc. It is an easy-to-use program that has been in operation a while.

The Color Computer version is a combination of Color Computer BASIC and Machine Language Programs. It also includes some special routines, such as BINCON/BAS, which allow the conversion of the TELEWRITER Word Processors' non-text files to be converted to text (or ASCII) files so they can be checked with this Program. Another example of the "marriage" to the Color Computer is the use of the BASIC Program SPELLFIX/BAS to provide the 'interface' between the Computer and the SPELLFIX/BIN Spelling Program, which provides both an easy operating system and the capability of making simple changes should your Computer configuration change in the future. Other routines supplied include: BUILD/BAS, a BASIC Program for generating short text files; LIST/BAS allows listing Text Files; RESET/BAS provides a "cold start" reset (used to restore pointers, etc., to allow other programs to be run after the use of SPELL 'N FIX without the need to turn the Computer Off, and back On, again); EXPAND/BAS allows the expansion of the "compressed" Dictionary file in case you wish to edit it. SPELL 'N FIX is almost worth the price just to have the BASIC Programs for study and reference.

A "spelling secession" with SPELL 'N FIX goes something like this:

LOAD "SPELLFIX" (or CLOAD, if Tape)
RUN

This loads the Program and execution begins. It asks if you want to use the Printer, then moves on to

ENTER FILE NAME OF TEXT FILE TO
BE PROOFREAD (NAME/EXT:DR)

At this time, the Program has been loaded, so you can switch Disks or Tapes, if needed, then tell it the File you want it to check. It then asks

WHAT KINDS OF WORDS DO YOU WANT?
A = ANY GROUP OF CHARACTERS ENCLOSED BY A
SPACE OR CR, OR
S = SELECT ONLY WORDS WHICH ARE REASONABLE
ENTER A OR S :

Here you select whether to check ALL character sequences or only those which are probably 'words' (for instance, skip format or printer control sequences). SPELL 'N FIX reads the Text File and arranges the words so they can be compared to the Dictionary, and lets you know it hasn't "taken a break" with

PROCESSING TEXT FILE ... PLEASE WAIT

When this is done, you get the message

SOURCE TEXT FILE HAS BEEN READ.
TOTAL NUMBER OF WORDS = 2345

ENTER FILE NAME (NAME/EXT:DR)
OF DICTIONARY FILE

The Text File has been read, and it wants to know what the name of the Dictionary is (keep it simple - most operating systems can't handle "WEBSTERS NEW WORLD DICTIONARY" as a Filename - an answer of DICT/DAT:0 will usually suffice, or just DICT). It then wants to know if you want to update the Dictionary by adding new words to it. This allows you to include words that you normally use that may not be in the normal Dictionary File, such as names, addresses, unique or specialized words, etc.

ARE YOU GOING TO BUILD A NEW
DICTIONARY FILE?
ANSWER Y OR N -

Again, it lets you know it's not goofing off with

PROCESSING DICTIONARY ... PLEASE WAIT

Each time it finds a word that is not in the Dictionary, it prints

MISTEAKS
I, M, X, OR Q -

You tell it to Ignore, Mark, X for MARK ALL from now on, or Quit to stop the embarrassment. This continues until all words have been checked (or you give up with a Q). Then you get a

DICTIONARY FILE IS FINISHED
BEING PROCESSED

The Marked words are then brought back and you choose

WOULD YOU LIKE TO WRITE A NEW
TEXT FILE WITH INCORRECT WORDS
EITHER CHANGED OR MARKED?
ANSWER Y OR N -

If you answer Y, it asks

WHICH — CHANGED OR MARKED?
ANSWER C OR M -

You are also asked again for the Text File name of the file that was checked, and you are asked for a new filename for the changed or marked file that will be output. If you "mark" the file, all words that were marked for checking will be followed by *** in the text file that is output, and can be located with "Find" or "Search" Commands within the Editor or Word Processor; or you can change the word as it comes up during the process of rewriting the file, producing a corrected Text File directly. If you are "changing", the preceding line is output to the Display Screen so you can see the context, and you enter the word as it is to be installed in the text (using upper and lower case, etc.). Finally, you get

SPELL 'N FIX IS FINISHED.

and back to the normal OK Prompt of BASIC.

This might have been a little long, but was presented for two reasons; 1. so you can see how SPELL 'N FIX Prompts for responses (by the time you have run a couple of text files through the Spelling Program, you will be at ease with it), and 2. so you can see how "Dictionary" type programs work.

The above example was oriented toward a DISK SYSTEM use of the Program - the Tape version is more restricted because you can not have more than one file open at any one time. With the Tape Program, the best procedure is to use the Printer output, so you can refer to it while updating the Text after it has been checked. Also, it is extremely difficult (I say this because NOTHING IS IMPOSSIBLE with Color Computer Users) to update the Dictionary with Tape.

BUT, THE GOOD NEWS for Tape Users

is that 1. SPELL 'N FIX on Tape is the same Program that is on Disk, and 2. instructions are provided for converting it over to Disk when you get a Disk System. This makes SPELL 'N FIX an excellent purchase even if you do not have a Disk System at the present time.

The Documentation that Pete supplies with the Program is the normal EXCELLENT presentation that owners of STAR-KITS products are used to seeing. Special notes, such as using the Program with different Word Processors, or changing the Program from Tape to Disk, make SPELL 'N FIX a program that will not be easily "outdated". If you are using the Color Computer for anything besides playing games on it, you need to add STAR-KITS SPELL 'N FIX to your stable of Programs.

QUICK LOOK REPORT:

DISK CONTROLLER BOARD
A Disk Controller, Real-Time Clock, Parallel Printer Adapter, and optional 80x24 Video Display Driver

ATOMTRONICS
SOUND CENTER RADIO SHACK
Whiterock Shopping Center
Los Alamos, N.M. 87544

without the Video Display, $274.95
with the Video Display, $349.95

We have a preliminary production Adapter Board for the TRS-80C Color Computer which is impressive, to say the least. ATOMTRONICS Disk Controller Board appears expensive until you see it and examine it's capabilities; WOW! The basic functions provided by the Board are a constantly operating Real Time Clock, a Double Density Disk Controller, and a Parallel output Printer connector. Optionaly, you can purchase the unit with a Video Display driver for an 80x24 Monitor Display with a Light Pen capability.

Physically, the PC Board is about the size of the shielded section of the Color Computer, and sets above the RF Shield Top Cover on the three plastic posts inside the Computer. It extends out to the back of the Top Cover of the Computer, where a slot approx. 1/2" by 6" is cut to allow connecting the Disk Cable and Parallel Printer Cable with Board Edge Connectors. A 2 to 3" long, 40 lead flat cable plugs into the ROMPAK connector from beneath the Top Cover, with a female connector mounted on the back of the connector to allow plugging ROMPAKS into the Slot for normal use. There are a couple of AA Pencells mounted to the Board for the Real Time Clock. This Board contains more components than is contained in the whole Color Computer, including a 6821 PIA, 6845 Video Display Controller, and 1793A Disk Controller chip; 2 24 pin ROMs, the new 24 pin MC146818 Clock Chip, and more than two dozen TTL support Chips. An RCA type connector provides the output for an 80x24 type Display Monitor, such as the NEC, AMDEX, SANYO, etc. units that take an NTSC standard input signal. A three pin Light Pen connector is also provided.

The Board was designed to be "adaptable" with numerous jumper configurations which allow such things operating the 1793 at 2MHz for 8" Disk Operations, configuring the Display Character Generator ROM for a 2732 for 256 characters capacity, changing the Disk Connector Pin numbers 6 and 32 to adapt to the various Disk Drives that interchange these two for DS3 Select or Side Select, etc. Software is provided for setting the Clock, the use of it's 50 bytes of CMOS RAM, etc. The Board also contains a special version of the Wolfbug Monitor in a 2732 ROM which includes a FLEX Compatible Boot Routine, and a couple of switches which allow the selection of Booting normally by picking up the FLEX Boot "Link" to FLEX.SYS, or pulling FLEX from Track 1, Sector 1 where it normally resides on the TSC General FLEX Disks. The other switch allows the display selection between the Monitor or TV Set. This function can also be performed thru Software, allowing "Changing Displays of the Fly", so to speak.

For the Color Computer Owner that wants to make serious use of the machine, the ATOMTRONICS Disk Controller Board may be the answer.

--- RLN ---

FMATE Usr Rout.

Steve Odneal
8609 East 73 Terrace
Kansas City, Missouri 64133

The purpose of this documentation is to provide the programmer with information which will allow usage of some of the special routines and data areas present in the F-MATE(R$)" FLEX" System Conversion for the Radio Shack TRS-80C" Color Computer as supplied by DATA-COMP".

Every effort has been made to insure that the information given is accurate for release 1.0, and every attempt will be made to keep these locations in later versions of DATA-COMP'S FLEX for the Color Computer. This assures Programmers who wish to develop software for the Color Computer/FLEX System that by using these entry points and memory locations, along with those documented for FLEX itself, their programs and other routines will be compatible with current and future releases of DATA-COMP'S FLEX for the Radio Shack Color Computer.

PRINTER ROUTINES

$E203 (RS2320) Serial Printer Output

This is an extended jump into a routine which provides for the output of data to an RS232 serial printer of the Color Computer.

This routine may be called via a 'JSR $E203' instruction. For example:

```
LDA  #'A    CHARACTER TO BE OUTPUT
JSR  $E203  PRINT THE CHARACTER
```

The status of the printer is checked before the character is output (using RS232S), and a loop is entered waiting for the printer 'Ready' line to become active before the serial data is output.

The ASCII character to be output to the printer must be in the A-register upon entry to this routine. All registers are restored upon exit, except for the condition codes.

The value in 'BAUD' at $E216 is used to determine the speed at which the character is serially output.

$E20F (RS232S) Printer Status Check

This is an extended jump into the routine which checks the status of the printer attached to the RS232 serial output port of the Color Computer.

This routine may be called via a 'JSR $E20F' instruction. For example:

```
JSR  $E20F  CHECK PRINTER STATUS
BMI  NOTRDY ..IF PRINTER NOT READY
```

All registers are restored upon exit, and there are no entry conditions. On exit, the Negative condition code bit is SET if the printer was NOT ready.

$E216 (BAUD) RS2320 Serial Baud-rate

This is a 2-byte value which specifies the RS232 serial printer output speed. The values which can be stored at this location cause the following serial speeds:

```
$01CA = 120 baud
$00BE = 300 baud
$0057 = 600 baud
$0029 = 1200 baud
$0012 = 2400 baud
```

This value has a default setting of $0057 (600 baud), and can be changed by the user at any time. For example:

```
LDX  #$0012  SET 2400 BAUD SPEED
STX  $E216
```

KEYBOARD INPUT

$E213 (INNO) Keyboard Input (no echo)

This is the ADDRESS or the routine which scans the Color Computer keyboard for input. The routine will wait for a key to be pressed before returning to the caller. A flashing cursor is displayed while waiting for input.

This routine may be called via a 'JSR [$E213]' (Indirect) instruction. For example:

```
JSR   [$E212]   WAIT FOR KEYBOARD INPUT
STA   CHAR      SAVE INPUT CHARACTER
```

All registers are restored upon exit from the routine, except for the A-register which will contain the Hex value of the character entered.

This routine will generate all available keyboard characters, including any User-Defined key values.

E21A (USERTAB) User-defined Key Values

This is a 2-byte address giving the start of a 12-byte table of values. This table contains the Hex values which will be generated whenever one of the corresponding 'user-keys' is input from the keyboard. The values correspond to Cntl-0 thru Cntl-9, Cntl-: and Cntl-;.

$E21C (POLCAT) Poll Keyboard

This is the ADDRESS of the routine which directly scans the keyboard of the Color Computer. The keyboard is scanned ONCE before returning to the user. No cursor is generated.

This routine may be called via a 'JSR [$E21C]' (Indirect) instruction. For example:

```
JSR   [$E21C]   SCAN KEYBOARD ONCE
BEQ   NOKEY     ..IF NO KEY PRESSED
```

There are no entry conditions, and all registers are restored upon exit from this routine, except for the A-register. If an input was made from the keyboard, the A-register will contain the Hex value generated by the key(s) pressed. Otherwise, the A-register will contain zero.

VIDEO INPUT/OUTPUT

$E400 (VIDINT) Video Initialization

This is an extended jump into the routine which will initialize the video display. The Color Computer 6883 SAM circuit is set for the proper video page and resolution, and the video screen is cleared. Control is returned to FLEX "WARMS", not the caller.

This routine may be called via a 'JSR $E400' instruction. For example:

```
JSR   $E400   INITIALIZE VIDEO DISPLAY
```

There are no entry conditions for this routine, and the A, B and X registers are DESTROYED upon exit.

$E403 (DISP) Display Character

This is an extended jump into the routine which will display a character on the video screen at the current cursor location. Values less than an ASCII space ($20) are ignored, and any parity-bit is stripped-off.

This routine may be called via a 'JSR $E403' instruction. For example:

```
LDA   #'B     CHARACTER TO BE DISPLAYED
JSR   $E403   DISPLAY THE CHARACTER
```

On entry, the character to be displayed must be in the A-register. All registers are restored upon exit, except for the condition-codes.

$E406 (VIDLIT) Video Format Literal

The 5 bytes at this location contain an ASCII value which defines in what format the video display is presently displaying characters.

The first 2 bytes give the number of characters per line - 32, 42, 51, 64, etc. The next byte ($E408) is a period. The last 2 bytes ($E409-A) give the number of lines which will be displayed on the screen - 16, 24, etc.

For exemple - If the current display format is 51 by 24, the literal value in these bytes will be '51.24'.

$E40B (CURSOR) Cursor Address

These two bytes can be used to determine the current location of the cursor on the video display screen.

For the standard 32x16 display, these bytes give the actual memory location where the next character to be displayed will be located.

However, for any of the high-resolution video display modes, these two bytes give the memory address of the BEGINNING of the current display LINE. The COLUMN and BIT values described below are also used to precisely define the memory location that will contain the next video character.

$E40D (COLUMN) Next Video Column Number

This byte contains the column number of the next character to be displayed when using any of the high-resolution video display formats. It can range from 0 to one less than the total number of characters which can be contained on one video line. It is used in conjunction with CURSOR and BIT to determine where to begin setting bits on or off to display the high-resolution character.

$E416 (BIT) Video Byte Bit-Number

This byte will contain a bit number corresponding to the first bit in COLUMN above which will be set on or off to display the required high-resolution video character. It is internally calculated by the video routine, based on COLUMN, and is documented here for completeness only. It may not be changed by the programmer.

DISK INPUT/OUTPUT

$DE4B (DRVTRK) Last Accessed Track

This is 4 bytes of information giving the binary number of the last track accessed for each of the four supported disk drives. It should not ba changed by the programmer, but can be tested for any purpose.

The bytes correspond to drive numbers 0, 1, 2 and 3 respectively.

$DE4F (DNSITY) Disk Density

This is 4 bytes of information which will give the density of the disk sector just previously read or written for each of the four supported disk drives.

A zero in a byte indicates single-density, and $FF indicates double-density. The disk routines will use the byte corresponding to the selected drive number to determine which density is to be used to read or write the next sector of data on the disk. The values should not be changed by the programmer, but can be tested for any purpose.

The bytes correspond to drive numbers 0, 1, 2 and 3 respectively.

$DE5B (DSPEED) Drive Seek Speed

These 4 bytes are used to determine the seek time used for each of the four supported disk drives.

As a default, each byte has an initial value of $03, which sets a seek time of 30ms for the Western Digital 1793 disk controller chip. Different Seek times can be set for faster drives. For example:

$00 = 3 milli-seconds
$01 = 6 milli-seconds
$02 = 20 milli-seconds
$03 = 30 milli-seconds

The bytes correspond to drive numbers 0, 1, 2 and 3 respectively.

SWTPC Meeting

SWTPC ANNUAL DISTRIBUTORS MEETING

For the second year in a row, Southwest Technical Products, Inc. held their annual distributor sales meeting, in San Antonio, Texas, April 23, 24, 25 and 26.

This years meeting was somewhat larger than last years with over 175 persons in attendance. It seems that the SWTPC distributor network has grown about 25 percent since last year, if attendance is any indication. This speaks well for our industry and SWTPC in these rather sparse economical times.

A short run down of some of the exhibits and applications software shown by various vendors including SWTPC is in order. On Friday the 23d we arrived and were entertained by SWTPC to a poolside buffet dinner at the Ramada Airport Inn. After a long two days drive we hit the sack early that evening, after enjoying the renewal of a lot of old friendships and making of some new ones.

Saturday got underway with a Continental Breakfast on the patio and then went directly into an open meeting. Gary Kay, SWTPC design engineer gave a look and talk on some of the new SWTPC products. After a short coffee break SWTPC Marketing Manager Fred Fuchs gave an overview of new SWTPC marketing and maintenance support programs to be shortly announced by SWTPC. This led to the introduction of Mel Shapiro Vice President of CARTERFONE, who will soon be offering nationwide service on all SWTPC products. To my knowledge this is the first of the Standard S50 Bus manufacturers who will have available nationwide service, on 24 hour call. Time will tell how successful this will be, but it is my opinion that they are going in the right direction and operational upgrades, such as this indicates a maturing of our industry.

Following lunch on the patio Fred Calev gave an impressive presentation of the varied UniFLEX™ applications software that they have developed at COMPUWORLD for the growing number of UniFLEX users. We here at 68 Micro Journal have tested a part of this software (Payroll) and if the rest of the line works as well and hassle-free as Payroll does, they have a winner.

Dan Vanada, Vice President of TSC gave an overview of current and coming UniFLEX upgrades and fixes to some past problems. Of special interest to most was the subject of maintenance for UniFLEX, coming utilities and seemingly most important a configurable version of the UniFLEX multi-user disk operating system. It is my opinion that the configurable version is long past due and needed if we are to continue to maintain a positive growth curve. It was agreed by most that more time should have been devoted to this portion of the meeting. Also there from TSC was the new marketing manager Don Sinklewicz, we all wish you well Don and hurry up and help those guys get the backlog caught up.

After a short coffee break (note I sure enjoyed the goodies breaks), Marion Chapman, a most lovely and personable lady, who is marketing manager for COMPUWORLD gave an enlightening discussion of the sales tactics they use and how they could benefit other distributors. It is certainly impressive to see these ladies who have involved themselves so well in the Standard S50 Bus community.

Then we were delighted again by another lovely and talented lady, Barbara Rush. I have known of Barbaras' fine software offerings for some years, however, I hope that Barbara makes it available to other distributors as her FLEX™ software comes well recommended. Again lovely ladies and software that works, make a difficult combination to beat. Barbara has much, time proven, business applications software running in quite a few different installations down in Texas, check with SWTPC about her software availability.

Fred Pfahler of Meta Lab was there showing their new CP/M hard/software. It was quite impressive and it appears that this could open the door to those Standard S50 Bus users who would like to take advantage of some of the CP/M software available. We will have a review started soon on their Z809 board and will try to let you know just how much value it could be to your system.

Surprisingly this was now Saturday evening and most everyone was still in attendance, there was a lot more to come.

Sunday started off again with a continental breakfast on the patio and then on with the meeting. Jerry O'Brien SWTPC Software manager laid out a vertical marketing sales approach and it appears that a lot of homework has been accomplished since last year as this was well received. It was about here that we learned of the new SWTPC warranty policy, to be soon announced. Seems that the warranty is going to be 90 days from date of delivery to the end user. The old policy of 6 months will probably be a thing of the past. For the distributor this is a better plan as it protects his shelf stock, whereas before it was possible for some slower items, stocked by some dealers who carry large inventories, to have the warranty period expire before delivery to an end user. Also all new systems and boards will be serial numbered. These policies along with the Carterfone national maintenance gives SWTPC a warranty program that is, or should be, clearly defined. This is an improvement over what existed in the past. Again a sign of maturity, and something other manufacturers might consider.

Richard Schuh gave a presentation on vertical farm marketing, interested parties should contact either SWTPC or one of their distributors for additional information.

A well received and extensive software package was presented by Gary Higbee of DP Systems of Chattanooga, Tennessee. We are very well acquainted with Gary, DP Systems and their medical software package. This software, running under UniFLEX is one of the most extensive projects I am aware of. I know personally that it took a couple of man-years to develop and has been running now for over a year in some large medical establishments. This is a very well done package and has features not normally found on micro-computer installations. It is a fine example of an excellent multi-tasking-user micro system and advanced software applications.

Maurice Leatherbury presented his Library Systems software package, interested users should contact SWTPC concerning this specialized package.

Chris Dickson, of England demonstrated a software programmer called GENESIS. Genesis is programmed in plain language by very inexperienced users and produces elegant code. This was a most interesting package, and its price also. Although somewhat expensive (could get near the three grand point) it is certainly less expensive than an in-house full time programmer. As the ole saying goes it don't grumble for raises and never goes on strike or take a coffee break. We were promised a review copy soon and we will report more on this after we have had ample opportunity to evaluate Genesis to a little more depth than was demonstrated there.

Russell Brown (old high slider Brown (motorcycle riders know what this is)) demonstrated a package that does some very impressive repair work to bombed hard disk under the UniFLEX system. We had an occasion, while in San Antonio to put this package to very good use. Seems one of our customers, a large pharmaceutical house here in Hixson had bombed their hard disk and had not backed off some very important data. My people flew the disk into SWTPC while I was there, for repair (disk that is), they asked us (actually nearly begged) to please try to repair the disk system and not loose the data, before practically an impossible situation. I relayed this plea to Fred Fuchs, Jerry O'Brien and George Abbott, Service Manager for SWTPC. Before I arrived back here in Hixson, the disk had been repaired by George and the data SAVED using the utilities Russell had just brought over from SWTPC England. This one has to be a winner. We were promised a review copy of this fine package soon. Watch for it!

Joe Gabay gave a talk on CPU-CPU Communication, and also demonstrated some interesting software.

Joe Turner of COMPUTER SYSTEMS CENTER demonstrated a very impressive software package running under FLEX (soon UniFLEX also) called DYNACALC. Dynacalc is an improved product over all of the other ---CALC type programs I have seen running on some other CPU systems. It sometimes still astonishes me how some folks can develop software and get it up and running without any major bugs and others seem to be continually mired in patches, fixes and 'hope I got it now' situations. Joe had just received this from its author and it was running first class at his demo. It did everything I asked it to and still had more tricks up its sleeve than you can shake a big (or even little) stick at. We will also have a review of this coming and needless to say, I was impressed. It is available NOW from Computer Systems Center in the FLEX version. Not confirmed was that there would also soon be a UniFLEX version of their DYNAMITE, a lot of programmers were certainly interested in this rumor. Let you know later.

Crite Wylie was also there demonstrating his SWTPC computer system driving and being driven by a well known cash register. This is a very important product for the average distributor, however, I did not have sufficient time to look this over well and suggest you contact SWTPC if you have need for this type application.

Joel Heckman of UNIVERSAL DATA RESEARCH was there with his 'Data Base Management System' running under UniFLEX, but Joel was called away out of town before he had an opportunity to present the features of his latest offering on UniFLEX. I know of many satisfied users and would suggest that you contact him or any of their dealers listed in their advertising, for more information.

Jim Carter explained his Service Business Management software running also under UniFLEX. Jim is an old customer and I personally am aware of the care in which he treats his products. We hope to have a review or user report on this also in the not to far future.

That afternoon we had a 'Show and Tell' open house where we all could gather around 8 to 10 SWTPC machines and see and try any or all of the above. Needless to say there just was too much to truly do justice to all. This lasted until past 11 p.m. and still I could not get around to all.

Monday was a wrap up half day and an open meeting was held between SWTPC reps and many of the distributors. I hope that some of the ideas laid out at this and other meetings is considered on both sides. We are in a period where a lot of 'shaking out' will be done. It seems that SWTPC has a handle on the future. That is good for all of us who use the Standard S50 Bus.

My overall impressions were that we have made some necessary and gainful strides in the past year. The introduction of a firm and understandable distributor warranty policy, a national service organization such as Carterfone is certainly a step forward, not to mention the new S+ system. There were a lot of new faces and names this year and also there were some old ones missing. Overall I believe that their distributor network has grown, which probably means that things are going pretty good. The economy being what it is, I trust that by late summer things should be on an up-beat trend for all of us.

Last but certainly not least was the introduction of the new SWTPC S+ concept. The new cabinets and advanced technology systems were beautiful both to the eye and technical appreciation. The S+ will give a lot of 'high-priced' minis a real run for their money. It looks very impressive (charcoal, eggshell white and a nice blue trim). This is a system that will blend in very well with any office decor. The rack mount is well thought out and it is in my opinion, both from the looks as well as functional angle, a most impressive computer system. With its many new features and good looks it seems light-years ahead of last years model. Also shown was a two piece CRT terminal, a 12 inch screen and detachable keyboard, very neat and professional looking. All of the new hardware I saw showed a lot of market research and forethought. Good looks and lots of options never hurt.

I would also like to thank Helen Meyer for the time spent entertaining my better half while I was busy snooping around, even if I did end up with a van full of packages coming back. Also Joyce and I want to thank Fred and Marlann Fuchs for a most delightful evening, in their lovely home with Helen and Dan Meyer. The steaks were delicious and of course the relaxed atmosphere poolside made a perfect ending to a busy but enlightening weekend. Tell you more next year, so keep tuned in.

DMW - - -

Color Clinic

TONY DISTEFANO
4680-18 Street
Laval West QUE
Canada H7R 2P9

I bought my first 4K Regular Basic Color Computer January 1980. On that same day I looked inside to see what made it tick. Since then I haven't looked back. Today I have expanded a 64K Disk Extended Basic Color Computer and my options are unlimited.

Hi there, my name is Tony DiStefano. What I would like to do in this column is to talk to you about my experience working with the Color Computer and to share the knowledge I have acquired. No matter if you're a hobbiest or a hardware hacker if you have any problems or questions concerning your computer, I will be glad to help you. I will answer all letters which have a SASE personally. All intersting letters will appear in this column. Your letters can be sent to "68 MICRO" or addressed to me. Enough of this, let's get into something more interesting.

After opening the computer, I noticed that all the ICs were made by Motorola. I immediately called my local Motorola representative and asked him for the data sheets on all the ICs that were in the computer. These data sheets are the backbone to understanding the computer. Anyone can get these sheets by contacting your local Motorola rep. I suggest that if you don't have them, go out and get them.

In this article I would like to clear up the controversy of the so called "HIGH SPEED" in the computer. As most of us know Poke 65495,"any number" speeds up basic programs by about 35%. But why does it work on some computers and not on others. Well! Here's the story.

All timings in the computer are derived from a 14.31818 mhz crystal. This frequency is made to input to the 6883 (SAM) chip. On power up, the Basic power up routine sets the SAM to divide the crystal frequency by 16, making the

6809E (MPU) frequency of .894 mhz clock rate. A write to $FFD7 ("$" Denotes hex number) or Poke 65495,0 sets the Sam into what is known as the A.D. (Address Dependent) rate. This means that the MPU will work only at one of two speeds .894 mhz or 1.788 mhz clock rates and this depending upon where the MPU is addressing. That's right! The SAM will switch between fast and slow clock rates depending where it is addressing memory. If the MPU is addressing memory between locations 0 and $7FFF (reading or writing) it will run at the slow clock rate. This area is ram. When it addresses memory between $8000 and $FEFF it will run at the fast clock rate. This area of memory is usually occupied by EXT BASIC, BASIC and DOS roms. I say usually because in another SAM mode this area could be ram also. In the I/O (input/output) area, any addressing done between $FF00 and $FF1F is at slow clock speed. The rest of the I/O area between $FF20 and $FFFF is at fast speed. This means that only one of the PIA's go to high speed. And not both like many people think. The PIA that does go to high speed is the one that does the D to A conversions and the VDG controls.

What does all this mean to you? Well you can use this information to find out why your computer doesn't work at the duel or high speed. We'll start with the easiest and least expensive ways. First if you have a disk drive, disconnect it, and try to get the computer to work alone. If the high speed doesn't work without the drive plugged in you will have to open the computer. (P.S. Refer to your service manual for instructions before you attempt to open your computer. Oh! by the way you may void your warranty.) Now remove the RF shield and locate the two capacitors C73 and C75. These two capacitors along with resistors R73 and R74 make up a RF suppression circuit. This unfortunately distorts the square wave shape of the E and Q clock signals. This prevents the system from working at the higher speed. OK, now turn the computer off, and remember to turn it off every time before you do a modification.

Cut one side of both capacitors. Why only one side? Because you may want to resolder it if it has no effect on the high speed after you cut it. Turn the computer on and try the high speed. If it works, great, if not you will have to continue. The next step is to check the PIA's. Since only one of the PIA's goes to high speed, the D to A and the VDG control one, try changing the PIA's arround. The chance that both will not work at high speed is rare. If the other one works then you are on your way. If not, well you will have to go one step further. At this point you may have to change some ICs. If you can, borrow rather than buy one 2 mhz PIA (MC 68B21) and one 2 mhz CPU (MC 68B09E) IC, because if after you have changed these two parts you may still be out of luck. Replace the CPU and the PIA with faster ones. Now it should work at the higher speed. If not, the only other components that you can change then are the Basic and EXT Basic ICs themselves.

With your computer working at high speeds it's time to try it with your Disk drive. What! It still doesn't work. Don't despair I have another trick up my sleeve. There is one more capacitor to cut, it is C85. This capacitor has the same purpose as the other two, RF suppression. Try the high speed with the Disk controller in. WOW! it works. And, if it should happen that your computer still doesn't work, the DOS rom may not be fast enough.

Chances are your system will now work at the higher speed. If you still have problems after cutting these three capacitors and changing the PIA and CPU drop me a line and describe the symptoms after you poke to high speed. I can make an analysis of the situation and prescribe a remedy. Well that's enough for this month. Good luck with "HIGH SPEED" and I'll see you next month.

OS-9 Seminar

Microware Systems Corporation is a privately held corporation based in Des Moines, Iowa, where it was established in 1976. Microware specializes in developing and marketing operating systems and programming languages for Motorola family microprocessors. The core of the Microware product line is a Unix-based multitasking operating system called "OS-9", which was developed during 1979 and 1980 in cooperation with Motorola, Inc. OS-9 has thousands of users in the areas of data processing, industrial automation, instrumentation, military systems, education, personal computers, robots, electronic communications and research. At the present time, products cover the 6809 advanced 8 bit MPU, and in the future will extend to the 68000.

Microware System Corporation's strategy is to produce versatile, state-of-the-art operating systems backed with high-level language support. Languages presently offered by Microware include, Basic, Pascal, Cobol, C, and Assembler. The firm also offers support software such as utilities, an editor, and an assembler.

Des Moines, Iowa temporarily held the title "THE WORLD'S MICROCOMPUTER SOFTWARE CAPITOL" as computer experts from all over the U. S. , Europe, and Japan converged on the city in the month of May. The reason for this unusual event happening in such an unlikely place was a seminar held May 14-16 at the Des Moines Marriott by Microware Corporation.

The engineers and scientists attending the seminar heard guest speakers, who are leading figures in the microcomputer revolution, and attended a day-long series of technical classes with such obtuse titles as "Real-Time Applications of Operating Systems". They also viewed exhibits of the latest microelectronic goodies displayed by about a dozen computer hardware and software manufacturers.

Large companies such as Western Electric, General Motors, Eastman Kodak, and Ford Motor Company were represented, as well as a number of smaller firms that specialize in particular aspects of commercial applications of microcomputer technology. One thing they all had in common is that they all are Microware customers.

"An important reason for this meeting was to establish a personal and professional basis for the exchange of ideas and technology between users of software", said Microware president Kenneth Kaplan. "Many times research and development efforts are duplicated by our customers. This meeting provided the foundation for a formal user's organization which can avoid such redundancy. There is so much to do and it takes too long for the latest technology to be applied to real products. We wanted to help our customers move as fast as possible in their rapidly changing markets."

Microware's only business is research and development of sophisticated computer programming languages, and operating systems. They are then licensed to computer manufacturers, who resell the software with their products, often as if they had developed the software themselves. Microware software products are used in tremendous variety of products ranging from Radio Shack personal computers to computerized blood analysis machines.

Why do Fortune 500 companies buy software from a small company like Microware? Charles Ball, Microware's marketing manager says that "This kind of research and development is best done by small teams of two or three people, and large corporations seem to have difficulty in attracting, retaining and managing the scarce few experts who know how to do these things. The problem is particularly acute overseas. Over 30% of our sales are outside the U. S. , especially Japan and Europe."

The following is an outline of the technical classes attended.

OS-9 System Technical Discussion
Larry Crane, Bob Doggett, Ken Kaplan
An overview and discussion of the organization and operation of OS-9 Levels One and Two.

OS-9 Input/Output System Seminar
Larry Crane, Bob Doggett, Bill Phelps, Bob Sorenson
A discussion of the OS-9 I/O system including operation of file managers, also using and writing I/O device drivers.

Software Marketing Seminar
Andy Ball
An overview of OS-9 features and advantages in the marketplace, developing a sales strategy, and other topics of interest to marketeers.

Selecting the Right Language
Ken Kaplan
A comparison of the OS-9 programming languages and how to match language features with application requirements.

Basic09 Advanced Topics
Bill Phelps, Larry Crane, Bob Doggett
Basic09 data structures, I/O functions, and structured programming techniques - Using RunB for ROMed systems.

Pascal Language Discussion
Mike Wimble, Steve Berens
An overview of the components of OS-9 Pascal and their use.

C Language Discussion
Dan Roady, Larry Crane
An overview of the C programming language and its implementation for OS-9.

Process Control and Real-Time Systems
Bob Sorenson, Larry Crane, Ken Kaplan
How to use OS-9 in control applications, especially small ROM-based systems Using Interrupts and Specialized I/O.

Business Software Under OS-9
Bob Doggett, Lewis Hibbits
Description of CIS COBOL, and business software written in Basic09 and CIS COBOL.

All this took place in one day (Saturday), with the seminar concluding on Sunday with a Brunch and Guest Speeches by Terry Ritter (father of the 6809 processor) and Larry E. Williams (Executive Editor 68' Micro Journal). Also on Sunday there was a Software Exchange Organizational Meeting conducted by Brian Capouch. (See Bit Bucket for details)

Sincere thanks to Ken Kaplan and the rest of the Microware staff for having been brave enough to hold this type of seminar. The general consensus of the people I talked with at the seminar and those to whom I spoke with over the phone after returning to my office was that the seminar gave them a great opportunity to get close to the people at Microware and ask questions in detail as well as the general aspects of OS-9. The atmosphere of cooperation and success of the classes made this seminar an event which will be eagerly awaited again next year. Special applause must be given to Jeanne Tunis for arranging the technical details as to the preparation which make seminars run smoothly as this seminar did indeed.

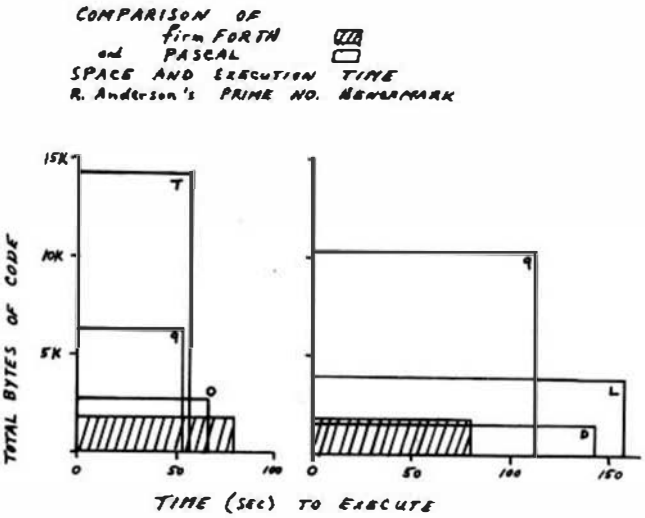
There is also available a Seminar Cassette Series from Microware at a cost of $25.00, including shipping and tax.

OOPS!!

Last month (June 82) we left out some charts that should have been included with the 'tFORTH' article on page 21. Sorry about that. We heard you loud and clear.

TABLE OF CHARACTERISTICS OF firmFORTH versus PASCAL

SYSTEM	# user bytes	Total # bytes	Time (sec) (1Mhz 6809)
firmFORTH	335	1788	80
OS-9 (native code)	919	6113	54
TSC "	721	14334	58
Omegasoft "	940	2465	66
OS-9 (p- code)	427	10241	112
Dynasoft "	301	1490	143
Lucidata "	598	3929	158



CC-Amateur Radio

BY: CLAYTON W. ABRAMS
1758 Comstock Lane
San Jose, CA 95124

The Radio Shack Color Computer is one of the best buys on the computer market today. Many of the people who have realized this are Amateur Radio operators. For many of you non hams it may be difficult to realize how computers may be used for this purpose, but hopefuly in this article I'll give you some insight.

BACKGROUND

Amateur Radio over the years have attracted a large number of creaters and experimenters. You would probably be suprised to find that many of the early ploneers in computers were hams.

To give you an idea of a few of the hams who are well known in the 68XX Micro world the numbers include Don Williams (68 Micro Journal), Bill Sias (CCN News), Bob Lentz (Microworks), and Dave Shirk (TSC). As always these early pioneer hams have used their talents in advancing the 68XX world of microcomputers.

Experimentation in Amateur Radio applications is unlimited. However, what you do in this special interest sideline can be shared by many people on a real time basis all over the world. Imagine sitting at your computer keyboard and banging keys, and having someone thousands of miles away printing your message. This is called Radio Teletype (RTTY). This can also be extended to the transmission of dots or dashes or commonly called Continuous Wave (CW). To go a step further the mode could be expanded to the transmission of television pictures or Slow Scan Television (SSTV). With the SSTV type of transmission television pictures can be transfered over HF communication frequencies or even telephone lines with a 8.3 second framing rate.

All of this is possible with computers, and I have been specializing in this special interest group since I constructed my first computer.

THE SYSTEM

My main interest in amateur radio is SSTV. In 1976 I decided to buy a computer system to replace my hardware SSTV character generator keyboard. At this time, components were very expensive, difficult to obtain,and there was almost no software available. Imagine paying $100 for a 4K RAM card ! I finally selected a SWTPC 6800 computer system and within 2 years I had the computer receiving and transmitting SSTV pictures and graphics (ref 1). The project was very rewarding, however it had two drawbacks.

1. The system required a lot of technical background to duplicate.
2. The system required extensive, expensive hardware.

So I continued to look for a low cost system that was easy to use and readily available. I set some goals for the project. I would consider a system only if the following criteria could be met:

1. The hardware had to be available anywhere in the world.
2. The hardware cost had to be as low as possible.
3. The computer must have the capability of displaying SSTV pictures.
4. The interfacing had to be simple and easy to build.

I investigated several different systems like, the Apple, the Atari, and the Pet.

The Apple had some of the necessary features, but its price was too high, besides it uses a 6502 processor UGH !

When Tandy announced their TRS Color with a list price of $399, I knew my search had ended. After obtaining a 4K unit in late 1980, I spent the first few months reverse engineering it, and found it was the ideal Ham Radio System because:

1. Unlike other TRS models this unit did not have TVI or RFI.
2. The TRS-80C uses the most advanced 16 bit (internal) processor available, the MC6809.
3. The unit has a built in A to D and D to A converter.
4. SSTV can be displayed on the computer with no revisions.
5. The system is easily expandable.

RTTY and CW

To understand how this unit is used in an amateur environment, lets explore the most common application for a ham radio computer, RTTY & CW. In these applications the computer requires an interface which has some signal processing. The receive interfaces are mainly used to remove noise. The transmit interface is used to provide the proper signals to the ham radio transmitter. In the computer all that is required is a single interface line which can turn off and on at a rapid rate. This line must be controlled by a computer program. In the TRS-80C the external hardware is connected by the RS-232 interface. Figure 1 contains a schematic of the interface inside the TRS-80C which generates and receives these signals. It is basicaly a level conversion op amp which applys the signal to a single bit of a 6821 PIA. The circuitry in the TRS-80C is rather tolerent and will accept either RS-232 level inputs or standard TTL signals.

If you attach a computer interface for CW and RTTY, this single bit under program control can be programmed to receive and transmit in either mode. With a little software it is possible to receive or transmit any code or bit pattern for this mode of transmission. To receive the code no external clock of baud rate is necessary. The clocking means can be a bit banger developed by program delays. In the TRS-80C a handy interrupt scheme is available for use. The horizontal video display sync pulses are fed into a PIA which is tied to the IRQ line. If you look for an interrupt every horizontal TV line you can count interrupts and develope accurate timings. This is the method used in my RTTYCW program.

The transmit interface for CW is a relay attached to the keying circuit of the transmitter. The transmit interface for RTTY is an AFSK modulator attached to the microphone input. The receive interface for both RTTY and CW is a demodulator which provides a two state input into the computer to decode. One of the interesting features added to the RTTYCW program is the facility to exchange programs over ham radio. I have exchanged BASIC programs over our local 2 Mtr calling frequency and over 20 Mtr RTTY ASCII 110 Baud with good results.

ANALOG INTERFACES

Probably the biggest asset of the TRS-80C is its ability to process analog signals. The Color Computer has a built in digital to analog (D/A) converter, and an analog to digital (A/D) interface.

The D/A interface in the computer is used to record programs on the tape recorder and is normally attached to the microphone input. Tandy uses this device to develope an audio signal to record casette program tapes. The D/A has 6 bits and the settling time is rather fast.

The A/D converter is used by Tandy for the joystick inputs to the computer. The A/D converter is a successive approximation type converter and uses the D/A to generate a test analog level. This level is placed into one leg of a comparator, the other leg of the comparator is connected to the analog signal. The output of the comparator is connected to a single bit of a PIA. With a little clever programming you can write a routine to digitize the analog signal. I was able to digitize an analog signal to 4 bit in approximately 50 micro seconds with the standard Tandy hardware.

To aid you in learning how these interfaces can be used, schematics are provided in figure 2.

SSTV GENERATION

You may wonder how these analog interfaces can be used for amateur radio? The answer is for amateur radio SSTV. Attached is the schematic of a simple SSTV modulator. With this modulator the color computer can be used to generate SSTV pictures and graphics. A schematic of this modulator is shown in figure 3.

In this mode the entire interface is controlled by software. You can generate pictures or graphics

entirely by software. Unlike hardware, software can be changed quickly to create video at almost any rate up to the limits of the processor.

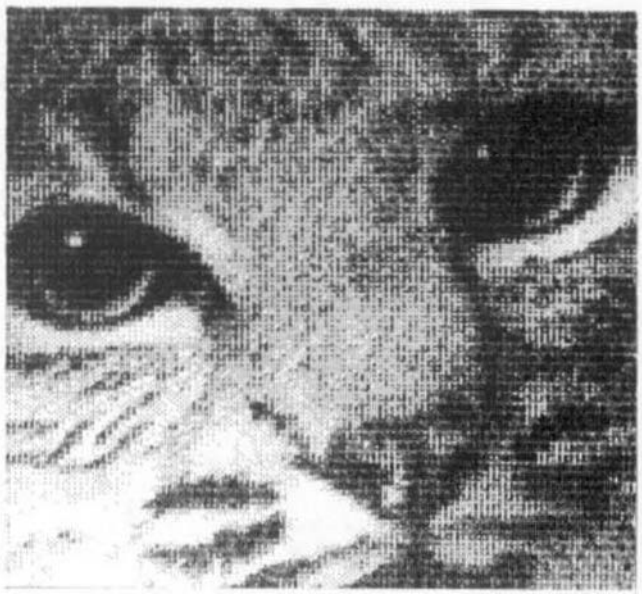
With this modulator the output from the D/A is used to develope the video. When the RS-232 line is dropped low, the interface outputs a sync frequency. When high the output of the SSTV modulator is controlled by the D/A. All that is required to generate a SSTV picture with this interface is a TRS-80C and a software package. The total parts cost is about $15, depending on your junk box. Everything is available mail order from a number of firms.

SSTV RECEIVE

If you can transmit SSTV the next trick is to display a SSTV picture on the TV attached to the computer. A number of firms offer hardware package to do this, but they all have a high price tag. Since many people have limited budgets for ham radio the TRS-80C makes a good compromise. Additionally, unlike the high priced hardware units the computer can be used for other things. I dont have much room in my ham shack and prefer to have multi purpose equipment.

I was very surprised by the quality of picture that can be displayed on this computer. In some cases the picture is as good as or better than a Robot 400. This unit is a commericaly available Amateur Radio Scan Converter and costs in excess of $600. In order to display pictures on the the TRS-80 two built in electronics devices are used in the computer.

One device is called the MC6847 which is a Color Display Generator IC manufactured by Motorola. The second IC is the MC6883 which is called the Synchronous Address Multiplexer IC or the SAM. The SAM chip is used to control the memory addressing of the computer. Under program control you can change the type of video displayed, the memory addressing of where the video memory is located and other functions. The MC6847 is the actual device which displays the TV picture. This device is also very flexable and can be initalized to fourteen modes of video display. These modes range from alphanumeric characters to graphics. I elected to use the graphics mode for SSTV. In this mode displays from 64 pixels on 64 lines to 256 pixels on 256 lines can be displayed with up to eight colors.



Now that you are aware that a SSTV picture can be displayed, lets see how you can get a picture into the computer. To do this you must first have some means of converting SSTV to analog and digital signals. The device that is used in all SSTV receivers is a frequency to DC voltage converter. It converts all input frequencys from 1500 HZ to 2300Hz to a DC voltage from 0 to 5 volts. If a frequency of 1200 Hz is present this voltage is converted to a sync pulse. In order to obtain these signals you can tap off a piece of SSTV equipment or build your own front end.

If you have a SSTV receiver the interface is very simple. It will work nicely with a Robot 400, Robot 70 or MXV 100 SSTV receivers. The schematic of the Robot 400 interface is shown in figure 4.

If you have no SSTV equipment the interface can be constructed for about $30.

This interface works by software control. To start the operation the computer must first wait for a vertical sync pulse, by polling the RS-232 input by software. When the pulse is sensed, the program samples the A/D converter to which the SSTV video is attached. As each pixel is received the appropriate value is placed in memory. Once a picture is in computer memory the possibilities are almost endless. Under programming control you can modify, zoom, analyze, or print pictures from the computer. You are only limited by your imagination and your ability to program.

SSTV SOFTWARE

As you must have guessed by now, the secret of using the TRS-80C for ham radio is the software. You will find that this is the critical link to the use of any computer system. Without the proper software the computer is as useful as a pet rock (silicon of course). To program a computer to adequately do all modes of ham radio takes a lot of work and equipment. I have been working on some of the programming techniques for SSTV for almost four years. Additionaly, to develope code for a large program requires a second larger computer system. In my largest SSTV program I have about 2500 lines of assembler language code, and the source listing is about 54 pages in length. My big computer system (6809) has 56K of RAM, three 5.25 disk drives, CT-82, Epson MX-80 and lots of specialized interfaces for ham radio.

SSTV RECEIVE SOFTWARE

Let's move on to give you more infomation on how the computer can be programmed to display a SSTV picture. The computer can be programmed to receive a picture in any one of 14 formats. I chose to use four formats in my SSTV7.4 package. However, only three of these formats were used to display pictures. The formats used were selected by a trial and error technique . Let's discuss the three modes by which the picture can be formatted:

1. High Density Mode- 128 pixels on 128 lines, 16 gray levels. This picture format is the same as the Robot 400. Using a 32K TRS-80C computer three pictures can be placed in memory. In this mode each pixel is four bits, and each byte is composed of two pixels. Since I can store three pictures the total memory required is 24K RAM. This allows me to use the remaining 8K for program. I did provide a feature in the program to quarter frame four pictures into one high density picture. Since the program allows pictures to be transmitted with a two times zoom, a total of 12 SSTV pictures can be stored in memory or on a casette tape, and transmitted one at a time. This format of picture density cannot be directly displayed on the TV attached to the computer. However, it can be directly transmitted over ham radio. I did provide a routine in the program to inspect the picture in a low density mode, to verify the picture which you are transmitting.

2. Receive Density- 128 pixels on 128 lines, or 128 pixels on 96 lines, 4 gray levels. In this mode a picture takes 4K of memory. In the program I can store up to four pictures in memory or on cassette tape. These pictures cannot be transmitted over ham radio. I did not provide this feature because the picture quality is somewhat reduced. You may wonder how a color display generator can display black and white pictures. It turns out if you disable the color on your TV, or you are using a black and white TV the 4 gray levels or colors become gray levels. For example yellow becomes white on a BW TV, blue becomes black and green and red become gray levels. For some reason some TV sets have difficulties resolveing red. I think it is a problem with the MC6847 IC. I tried using a 13 inch Zenith TV and the display of red was poor, but the RCA Color Tract works great. This is the set Tandy uses for their Color Monitor. For this display mode I used the 6C or 4C 6847 Graphics mode. I also found that a Color SSTV picture could be displayed. The quality is not as good as a three memory Robot 400 system. This program option only added about 200 bytes to the program. More on this later.

3. Low Density Mode- 64 Pixels on 64 lines, 4 gray levels. This mode was added to provide a quick inspection of a high density picture and used the 2C graphics mode of the 6847.

SSTV TRANSMIT SOFTWARE

Transmitting of a SSTV picture is a simple concept. A memory picture byte is first loaded into a accumulator and divided into two pixels. Each pixel is then stored into the A/D which causes the frequency to change on the SSTV modulator. You then delay a short time before sending the next pixel to the A/D. After 128 pixels are sent you switch the SSTV modulator to 1200Hz which is the sync frequency. You then delay for 5 or 50 milliseconds depending whether the pulse is horizontal or vertical. To do a two times zoom you transmit each pixel twice, and each line twice. This will give you a two times zoom on the transmission of the picture. Although the principal is quite simple, the software is critical. The correct number of microprocessor machine cycles must be counted, and even the addition or deletion of a single instruction may cause problems.

FIGURE 1

RS-232 INTERFACE (TRS-80C)

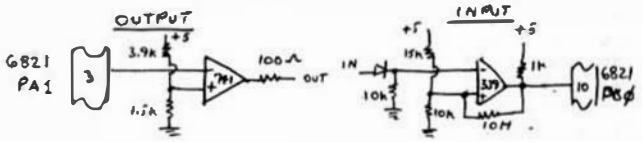
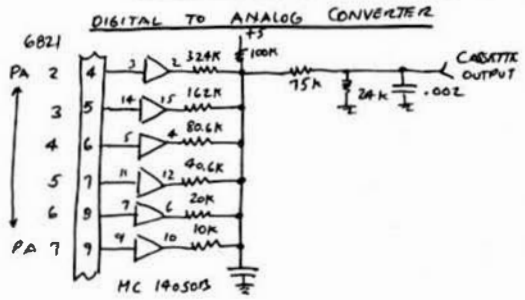
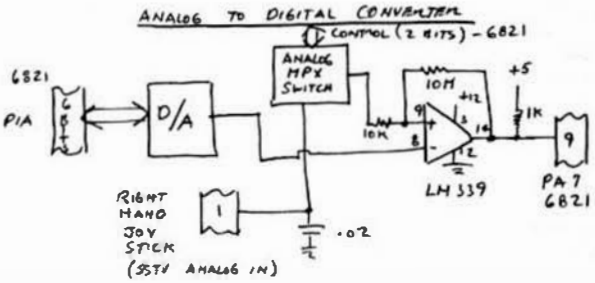


FIGURE 2

TRS-80C ANALOG INTERFACES





NOTE: 1. CONVERSION TIME APPROX 75 μSEC FOR 4 BITS USING SSTV 7.4 PROGRAM

2. PROGRAMMING TECHNIQUE - SET UP MPX AND OUTPUT TO D/A, LOOK IN PA7 HIGH. CORRECT VOLTAGE IS STORED BY TRIAL AND ERROR.

FIGURE 3

SSTV MODULATOR

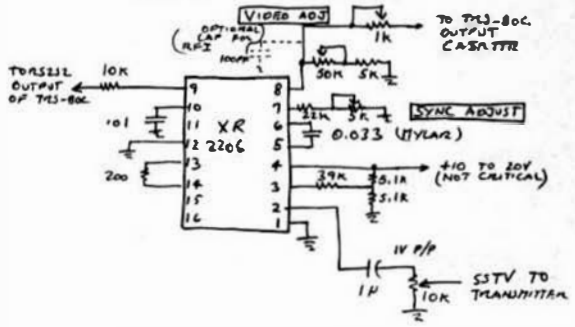
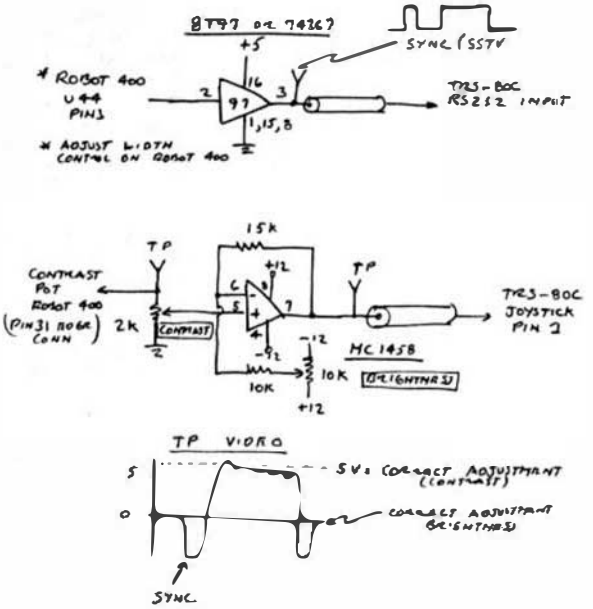


FIGURE 4

SSTV RECEIVE INTERFACE ATTACHMENT TO ROBOT 400



COLOR SSTV RECEIVE SOFTWARE

You may be wondering how color SSTV can be displayed on the computer? The current methods for transmission and reception have not changed very much for the last three years and are rather primitive. The transmission method is to send first a red frame then a green frame then a blue one. Most people who have the capability to display color receive each picture into one of three memories. Each memory is attached through an electronics multiplexer to create and analog TV signal on the individual guns of a TV picture tube. This method is very effective, but is costly to impliment. The method which I used with the TRS-80C requires no modifications to the computer or the TV set. The MC6847 in the computer takes care of all the formatting of the signals, and only a software algorithm is required to display a color SSTV picture. Lets see how this is done. The first step is to receive three SSTV pictures into memory in specific locations. The next step is to execute a computer program to look at each pixel in each location and combine them into a fourth location to form a picture. To do this you must understand how color SSTV works. When each color SSTV picture is loaded into memory, it is just a normal black and white picture. Each picture memory becomes a component of a composite picture. Lets take for example that you wish to display a red pixel on the TV. The red portion of the color picture will contain a white pixel, and the green and blue portions will contain black pixels. With a little clever programming you can scan three pictures in a few seconds and create a fourth color picture. When you switch to the 6C mode of the display, the computer will allow only 4 colors to be displayed. I must admit the method is very crude in this implimentation but the principals can be applied to other display IC's.

As you can see with a computer you are not restricted to a dead ended situation as can happen with the hardware scan converters. Your dollar investment is protected from future obselence by the ability to expand the computer as technology advances.

SSTV PICTURE PRINTING

A new and exciting side line of SSTV which is causing a lot of attention on the various amateur frequencys is picture printing. A number of the new generation of printers now being used on computers have a capability of printing pictures with a high resolution graphics mode. One of the most popular printer with this feature is the Epson MX-80. The Epson printer uses a dot matrix print head which is replaceable when worn out. A feature of this printer which was recientl announced is Graftrax. This feature is a slight microcode change in the printer which will allow an external program to access each wire of the print head and print a specific dot pattern. The Graftrax feature comes from Epson in three 2716 EPROMS and will replace the single ROM which is standard with the printer.

Two dot patterns are possible which display different densitys of print. The lowest pattern of graphics printing 480 dots per line, and the highest is 890 dots per line. Since it is possible to place a SSTV picture in memory with some external hardware and internal software, it would be an interesting exercise to print a picture. Attached is an example of a printed picture. This picture was taken from a TV camera. The TV camera was attached to a hardware SSTV scan converter which was constructed seven years ago long before home computing. The image was inputted to the TRS-80C Color computer by a hardware SSTV demodulator and digitized by my SSTV74 software package and printed by SSTV75.

CONCLUSIONS

The TRS-80C makes one of the best amateur radio home computers on the market to date. Its flexability and ease to program allow an easy integration into most ham stations.

More technical information on how a 68XX computer can be used for Amateur Radio SSTV is contained in a text book which was published last year (ref3).

If you are interested in obtaining more infomation on the software packages I have developed for the TRS-80C for RTTY,CW and SSTV drop me a line with a SASE or IRC and I'll provide a copy of listing of amateur radio software available.

73's Clay Abrams K6AEP

REFERENCES

1. SSTV Meets SWTPC- Parts I and 2, 73 Magazine, November and December 1978
By: Clayton W. Abrams

2. Amateur Radio and the TRS-80C Color Computer, A5 Magazine, Jan,Feb 1982
By: Clayton W. Abrams

3. Microprocessor Applications Handbook, Mc Graw Hill 1981, Edited by:
David Stout, Chapter 15, A SSTV System Using A Microprocessor, by:
Clayton W. Abrams.

Home Acct Prog

THE WATSON/BRADY HOME ACCOUNTING PROGRAM SYSTEM

I.
FOREWORD

A. REQUIREMENTS FOR USE OF THE SYSTEM

USE OF THIS HOME OR SMALL BUSINESS ACCOUNTING SYSTEM REQUIRES THE FOLLOWING:

1. BASIC KNOWLEDGE OF DOUBLE-ENTRY BOOKKEEPING;
2. SOME BASIC PROGRAMMING EXPERIENCE (TO INITIALIZE THE SYSTEM);
3. TSC'S EXTENDED BASIC RUNNING UNDER EITHER FLEX2 OR FLEX9;
4. SOME OF TSC'S FLEX UTILITIES AS INDICATED BELOW; AND
5. AT LEAST TWO 5" DISK DRIVES (OR THEIR EQUIVALENT IN CAPACITY) PLUS SUFFICIENT MEMORY TO RUN THE PROGRAMS AND FLEX.

IF YOU DO NOT HAVE, OR CANNOT MEET THE ABOVE QUALIFICATIONS, PLEASE DON'T ATTEMPT TO USE THIS SYSTEM. IT IS NOT A SUBSTITUTE FOR EITHER A KNOWLEDGE OF ACCOUNTING OR ADEQUATE EQUIPMENT. IF YOU WISH TO ATTEMPT TO ADAPT THIS SYSTEM (OR ITS PREDECESSOR SEQUENTIAL FILE SYSTEM) TO A CASSETTE-BASED SYSTEM, THE AUTHORS WOULD BE GLAD TO FURNISH USERS WITH COPIES OF THE PREDECESSOR XBASIC AND CASSETTE BASED PROGRAMS FOR THAT PURPOSE. THE AUTHORS WOULD ALSO BE PLEASED TO RECEIVE COMMENTS ABOUT, AND SUGGESTED MODIFICATIONS TO, THE SYSTEM. YOU MAY ADDRESS CORRESPONDENCE TO:

ERNEST STEVE WATSON
11701 ST. CHARLES BLVD.
LITTLE ROCK, ARKANSAS 72211

OR

F. DALE BRADY
7729 BRADLEY DRIVE
LITTLE ROCK, ARKANSAS 72209

B. FUTURE DEVELOPMENTS FOR THE SYSTEM

1. ONE OF THE PROBLEMS INHERENT IN THE USE OF THE SYSTEM IS THAT COMPOUND JOURNAL ENTRIES (BOOKKEEPING ENTRIES HAVING MORE THAN A SINGLE DEBIT AND A SINGLE CREDIT ENTRY) CANNOT BE MADE, AT THE PRESENT TIME. THIS IS NOT A SERIOUS HANDICAP, BUT IS ANNOYING. THE AUTHORS WOULD BE PLEASED TO RECEIVE SUGGESTIONS CONCERNING THE INTRODUCTION OF THIS FEATURE INTO THE SYSTEM. SUGGESTED MODIFICATIONS WILL BE SENT TO '68' MICRO JOURNAL FOR PUBLICATION IF THERE IS SUFFICIENT INTEREST IN THE W/B HAP SYSTEM.

2. SOME OF THE ENCLOSED PROGRAMS HAVE NOT BEEN FINALIZED, ALTHOUGH ALL OF THE PROGRAMS ESSENTIAL TO THE FUNCTIONING OF THE SYSTEM ARE COMPLETE. THE AUTHORS HOPE THAT USERS WILL BE SUFFICIENTLY INTERESTED IN THE SYSTEM TO HELP THEM FINISH IT. SOME PROGRAMS USERS CAN HELP WITH ARE: BALANCE.BAS AND BALANCE1.BAS. ALSO NEEDED IS A CHECKING ACCOUNT RECONCILIATION PROGRAM, AND, FOR SMALL BUSINESS USERS, AN ACCOUNTS RECEIVABLE, ACCOUNTS PAYABLE, AND PAYROLL PROGRAM. SINCE ADJUSTING AND CLOSING JOURNAL ENTRIES AT YEAR-END MUST NOW BE MADE MANUALLY, A PROGRAM TO CLEAR ALL INCOME AND EXPENSE ACCOUNTS AT YEAR-END WOULD ALSO BE A PLUS FOR THE HAP SYSTEM.

II.
GETTING THE SYSTEM STARTED

THE AUTHORS SUGGEST THE FOLLOWING IN GETTING THE SYSTEM UP AND RUNNING:

1. MAKE A BACK-UP COPY OF THE MASTER DISK FOR USE IN THE EVENT THAT YOU CRASH THE SYSTEM.

2. LIST OUT EACH OF THE PROGRAMS TO SEE WHAT EACH DOES AND TO BE SURE THAT ANY NON-STANDARD PORTIONS OF THE PROGRAMS WILL NOT CONFLICT WITH YOUR SYSTEM. NOTE THAT THE AUTHORS' PRINTERS REQUIRE THE POKE.CMD TO ACTIVATE THEIR PRINTERS. THIS COMMAND SHOULD BE DELETED AND YOUR PRINT COMMANDS SUBSTITUTED PRIOR TO RUNNING THE SYSTEM. NOTE ALSO THAT THE PROGRAMS PRINCIPALLY USE RANDOM FILES. THE ENCLOSED "GENLED.TXT" IS A SEQUENTIAL FILE INCLUDED FOR ILLUSTRATION ONLY AND WILL NOT BE SUITABLE FOR USE, AS IS, IN THE HAP SYSTEM.

3. PUT THE APPROPRIATE FLEX UTILITIES ON THE NEW MASTER COPY WHICH YOUR INSPECTION OF THE PROGRAMS INDICATES WILL BE CALLED OR USED. THE FOLLOWING FLEX COMMANDS WILL BE NEEDED: EXEC, PROT, TTYSET, LIST, FILES, PDEL (OR DELETE), COPY, CAT, AND ASN. THE AUTHORS, BECAUSE OF COPYRIGHT RESTRICTIONS, COULD NOT INCLUDE THESE FLEX UTILITIES WITH THE SYSTEM. MAKE CHANGES IN THE PROGRAMS AS NECESSARY FOR ANY SUBSTITUTES FOR THESE COMMANDS..

4. INITIALIZE A SECOND DISK FOR THE WORK DRIVE AND PUT THE NEW MASTER, WITH THE FLEX UTILITIES NOW ON IT, IN THE SYSTEM DRIVE, AFTER LOADING XBASIC. LOAD AND RUN "0.INTYEAR". DISREGARD THE ERROR MESSAGE. WHEN THE MENU COMES UP, SELECT "ADD NEW ACCOUNT NUMBERS" AND ENTER YOUR BEGINNING GENERAL LEDGER ACCOUNTS AND BALANCES. ASK FOR A SUFFICIENT NUMBER OF ACCOUNTS TO ENTER ALL ACCOUNTS NEEDED. INITIALLY, ALL CREDIT BALANCES SHOULD BE ENTERED AS MINUSES (-). DEBIT BALANCES NEED NOT BE ENTERED AS PLUSES (+). AFTER INITIALIZATION, THE SYSTEM WILL AUTOMATICALLY HANDLE ALL DEBITS AND CREDITS. "GENLED.TXT" HAS BEEN INCLUDED AS A SAMPLE GENERAL LEDGER. IT CAN NOT BE USED IN ITS PRESENT (SEQUENTIAL) FORM BUT CAN BE LISTED. BE SURE THAT YOUR GENERAL LEDGER IS IN BALANCE BEFORE ENTERING IT. THE SYSTEM WILL CHECK FOR THIS CONDITION AND ABORT IF OUT OF BALANCE. OUR CURRENT GENERAL LEDGER ACCOUNTS USE: "100" SERIES ACCOUNTS AS ASSETS, "200" SERIES AS LIABILITIES, "300" SERIES AS EQUITY, "400" SERIES AS EXPENSES, AND "500" SERIES AS INCOME ACCOUNTS.

6. USE THE SYSTEM EXPERIMENTALLY TO CREATE A SMALL TEST DECEMBER GENERAL LEDGER AND THEN A SMALL JANUARY TRANSACTION FILE AND JANUARY GENERAL LEDGER. IF ALL HAS GONE WELL WITH BOTH OF THESE, THEN YOU CAN CREATE A GENUINE DECEMBER GENERAL LEDGER AND SUBSEQUENT MONTHS GENERAL LEDGER AND TRANSACTIONS FILES.

7. IF ALL ELSE FAILS, WRITE THE AUTHORS AT THE ADDRESSES INDICATED ABOVE.

III.
SYSTEM SPECIFICATIONS

1. THE SYSTEM WILL HOLD ABOUT 850 RECORDS, WHEN USING TWO 5" DISK DRIVES. A PROPORTIONATELY LARGER NUMBER OF RECORDS CAN BE STORED AND ACCESSED IF 8" DRIVES ARE USED. FOR THE AVERAGE SMALL USER, THIS AMOUNT OF RECORDS SHOULD BE SUFFICIENT FOR AT LEAST ONE OR TWO YEARS. IF "TRAYEAR.BAS" IS MODIFIED TO DELETE EACH MONTHLY TRANSACTION FILE AS IT IS ADDED TO THE "YEAR.DAT" FILE, THEN TWICE THE AMOUNT OF RECORDS MAY BE STORED.

2. THE SYSTEM, IF MAINTAINED ON A MONTHLY BASIS, WILL PERMIT INSTANT RECALL OF ANY CHECK, DEPOSIT, OR SIMILAR ITEM ENTERED DURING THE CURRENT YEAR. IT IS INVALUABLE AT TAX RETURN TIME FOR SWIFT TABULATION OF AVAILABLE DEDUCTIONS, INCOME, ETC.

3. AS CURRENTLY STRUCTURED, THE SYSTEM CAUSES EACH MONTH'S TRANSACTION FILE TO BE CLOSED AND POSTED TO THE GENERAL LEDGER BEFORE ENTRIES FOR THE FOLLOWING MONTH

CAN BE ENTERED. THE SYSTEM ALSO MAINTAINS THE SEPARATE IDENTITY OF EACH MONTH'S TRANSACTION AND GENERAL LEDGER FILE, ALTHOUGH USERS DESIRING TO DELETE PAST MONTHS' TRANSACTIONS FILES MAY EASILY DO SO, BY CHANGING "TRAYEAR.BAS" AS INDICATED ABOVE.

```
0 REM PROGRAM WILL INTIALIZE YEAR FILE WITH YEAR
20 REM ,NAME OF COMPANY,DEC OLD GL MONTH AND
30 REM JAN AS CURRENT POSTING MONTH. SHOULD BE USED
40 REM ONLY AT BEGINNING OF YEAR.
50 REM THIS PROGRAM WILL NOT BE CALLED FROM MENU BUT
60 REM WILL RETURN TO MENU.BAS AT END
70 REM INTYEAR.BAS
75 EXEC,"0.ASN,S=A,W=1"
80 CL$=CHR$(27)+"E"
90 W=60
100 PRINTCL$
110 PRINTTAB(W/2-12);"Intialize Year Data File."
120 PRINT:PRINT
130 PRINT"This program should be used ONLY at
    beginning year."
140 PRINT"Enter END to return to MENU"
150 INPUT"Enter New Year ",AN$
160 IFAN$="END"THENCHAIN"MENU.BAS"
170 Y=VAL(AN$)
180 IFY<1980 OR Y>2000THENPRINTCHR$(7);"ENTER WHOLE
    YEAR (EXAMPLE '1990')":GOTO150
190 PRINT:PRINT"This will create a YEAR file for ";Y
200 PRINT"is this correct (Y/N)? ";:AN$=INCH$(0)
    :PRINT
210 IFAN$<>"Y"THEN100
220 PRINT:PRINT"Enter name of COMPANY";:INPUTLINEA$
230 Y$="1.YEAR"+MID$(STR$(Y-1),2,4)+".DAT"
240 PRINT"RENAMING CURRENT YEAR FILE TO ";Y$
250 E$="RENAME,I.YEAR.DAT,"+Y$
260 EXEC,E$
270 PRINT"PROTECTING OLD YEAR FILE"
280 E$="PROT,"+Y$+",WD"
290 EXEC,E$
300 OPEN NEW "1.YEAR" AS I
310 PRINT:PRINT"CREATING ";Y;" YEAR FILE FOR ";A$
320 FIELD#1,2ASTN$,2ASTY$,3ASTM$,3ASGL$,40ASN$
330 LSETTN$=CVT%$(0):REM NUMBER OF RECORDS IN YEAR
    FILE (0 NOW)
340 LSETTY$=CVT%$(INT(Y)):REM CURRENT YEAR
350 LSETTM$="JAN":REM CURRENT MONTH NOT POSTED
360 LSETGL$="DEC":REM LAST POSTING MONTH
370 LSETN$=A$
380 PUT#1
390 CHAIN "0.MENU.BAS"
400 REM 2 AS %= NUMBER OF RECORDS IN YEAR FILE
410 REM 2 AS %= CURRENT YEAR
420 REM 3 AS $= CURRENT MONTH NOT POSTED (TRANSACTION
    FILE)
430 REM 3 AS $= LAST POSTING MONTH
440 REM 40 AS $=NAME OF COMPANY

0 REM MENU.BAS
15 EXEC,"ASN,W=A,S=A"
20 W%=30:REM WIDTH/2
30 ONERROR GOTO 490
35 POKE44058,0:REM LINE COUNT
40 CL$=CHR$(27)+"E"
50 PRINTCL$
60 PRINT:FORX%=ITOW%*2:PRINT"*";:NEXTX%:PRINT:PRINT
70 PRINTTAB(W%-7);"WATSON/BRADY SYSTEM"
80 PRINT
90 READN$,P$:IFN$="0"THEN100 ELSE T%=T%+I:GOTO90
100 DIMI$(T%),P$(T%)
110 RESTORE
120 FORI%=ITOT%:READI$(I%),P$(I%):NEXTI%
130 FORI%=I TOT% STEP 2
140 IFI%+1>T% THENPRINTI%;". ";I$(I%):GOTO 170
150 PRINTSPC(I-I%/10);I%;". ";I$(I%);TAB(W%);SPC(I-
    (I%+1)/10);I%+1;". ";I$(I%+1)
160 NEXT I%
170 PRINT:FORX%=ITOW%*2:PRINT"*";:NEXTX%:PRINT:PRINT
180 PRINT
190 PRINT"Enter your choice? ";:AN$=INCH$(0)
200 S=VAL(AN$)
210 IFS>ITHEN240
220 AN$=INCH$(0):IFAN$=CHR$(13)THEN240
230 S=S*10+VAL(AN$)
240 PRINT
250 IFS=ITHENPRINT:PRINT"Thanks for using the
    WATSON/BRADY System";CHR$(7):END
260 IF S<1 OR S>T% THEN PRINTCHR$(7);"  NOT VALID
    CHOICE ":GOTO 190
270 PRINT:PRINT"Loading ";I$(S)
280 PRINT:PRINT"Please wait"
290 CHAIN P$(S)
300 DATA End program,xxxx
310 DATA Enter  Transactions,TENTRY.BAS
320 DATA Post Gen Ledger,POSTGL.BAS
330 DATA Search Transactions,TSEARCH.BAS
340 DATA Edit Gen Ledger,EDITGL.BAS
350 DATA List   Transactions,TLIST.BAS
360 DATA List OLD Gen Ledge ,OLDGL.BAS
370 DATA Edit   Transactions,TEDIT.BAS
380 DATA Budget Year,BUDGET.BAS
390 DATA Account Ledger,ACCLEDGR.BAS
400 DATA Check Writer,CHECKWTR.BAS
410 DATA Print Statement,PRTSTAT.BAS
420 DATA Check Book,CKBOOK.BAS
430 DATA Balance Sheet,BALANCE.BAS
440 DATA Bill 1st,BILLPAY.BAS
450 DATA Add NEW Acct.#,ADDGL.BAS
460 DATA Bill 15th,BILLPAY.BAS
470 DATA Delete Account #/NAME,DELGL.BAS
480 DATA 0,0
490 IFERR=4THENPRINT"OPTION NOT VALID AT THIS TIME"
    :PRINT:CHAIN"MENU.BAS"
500 ONERROR GOTO

0 REM POSTGL.BAS
20 CL$=CHR$(27)+"E":PRINTCL$
30 W%=60
40 PRINTTAB(W%/2-12);"Posting to General Ledger"
45 PRINT:PRINT"THIS PROGRAM IS USED TO POST
    TRANSACTIONS"
46 PRINT"TO LAST MONTH'S GENERAL LEDGER AND CREATE
    NEW"
47 PRINT"GENERAL LEDGER FILE WITH POSTED ACCOUNTS"
50 REM GET DATA FROM YEAR FILE Y%=YEAR,M$=CURRENT
    MONTH,P$=POSTING MONTH
60 OPENOLD"1.YEAR"AS1
70 GET#1,RECORD1
80 FIELD#1,2ASTN$,2ASCY$,3ASCM$,3ASPM$
90 Y%=CVT$%(CY$):M$=CM$:P$=PM$
100 CLOSE I
110 PRINT:PRINT"The ";M$;" transaction file will be
    posted to ";P$
120 PRINT:PRINT"Is this correct (Y/N)? ";:AN$=
    INCH$(0) :PRINT
130 IFAN$<>"Y"THENPRINTCHR$(7);"OK going back to
    MENU" :CHAIN"MENU.BAS"
140 PRINT:PRINT"Reading data from ";P$;" general
    ledger."
150 OPENOLD"1."+P$+".GL" AS I
160 SET#1,RECORD1
170 FIELD#1,2AS%$:X%=CVT$%(%$)
180 DIMN%(X%),A$(X%),A(X%)
190 GET#1,RECORD1:GOTO210
200 GET#I
210 FORS%=0TO7
220 FIELD#1,S%*30AS%$,2ASGN$,20ASGM$,8ASGT$
230 IFI%=0THEN260
240 N%(I%)=CVT$%(GN$):A$(I%)=GM$:A(I%)=CVT$F(GT$)
245 A=A+A(I%)
250 IFI%=X%THEN290
260 I%=I%+1
270 NEXTS%
280 GOTO200
290 CLOSE1
292 A=ABS(A):IF A>.01 THEN GOTO 295
293 GOTO 300
295 PRINTUSING"OLD GEN LEDGER OUT OF BALANCE
    $###,###.##",A:END
300 REM DATA FROM TRANS FILE
```

```
310 OPENOLD"I."+M$AS1
320 GET#1,RECORD1:FIELD#1,2ASZ$:TZ=CVT$%(Z$)
330 DIMDZ(TZ),CZ(TZ),AM(TZ)
340 PRINT:PRINT"Reading ";TZ;" transactions from ";
    M$;" file."
350 GET#1,RECORD1:GOTO370
360 GET#1
370 FORSZ=0TO4
380 FIELD#1,SZ#50ASZ$,2ASTD$,2ASTC$,2ASTN$,10ASCY$,
    26ASTP$,BASTA$
390 IFTP$="" OR CVT$%(TC$)=0 THEN430
400 JZ=JZ+1
410 DZ(JZ)=CVT$%(TD$):CZ(JZ)=CVT$%(TC$)
    :AM(JZ)=CVT$F(TA$)
420 IFJZ=TZTHEN440
430 NEXTSZ:GOTO360
440 CLOSE 1
450 PRINT:PRINT"POSTING  PLEASE WAIT"
460 FORIZ=1TOXZ:FORJZ=1TOTZ
470 IFNZ(IZ)=DZ(JZ)THENA(IZ)=A(IZ)+AM(JZ):A=A+AM(JZ)
480 IFNZ(IZ)=CZ(JZ)THENA(IZ)=A(IZ)-AM(JZ):A=A-AM(JZ)
490 NEXTJZ:NEXTIZ
500 IF INT(ABS(A$100))<>0THENPRINTUSING'ERROR POSTING
OUT OF BALANCE BY $###,###.##',A:PRINTCHR$(7):"HIT
ANY KEY TO RETURN TO MENU.";INCH$(0):CHAIN"MENU.BAS"
510 PRINT"POSTING IN BALANCE"
520 PRINT:PRINT"CREATING NEW GENERAL LEDGER FOR ";M$
530 ONERROR GOTO 720
540 OPENNEW"I."+M$+".GL" AS I
550 RZ=0
560 FORSZ=0TO7
570 FIELD#1,SZ#30ASZ$,2ASGN$,20ASGM$,BASGT$
580 LSETGN$=CVT$%(NZ(RZ+SZ))
590 LSETGM$=A$(RZ+SZ)
600 LSETGT$=CVTF$(A(RZ+SZ))
610 IFRZ+SZ=XZTHENPUT#1:GOTO650
620 NEXTSZ
630 PUT#1
640 RZ=RZ+8:GOTO560
650 GET#1,RECORD1:REM ADD FILE SIZE
660 FIELD#1,2ASZ$
670 LSETZ$=CVT$%(XZ)
6 0 PUT#1,RECORD1
690 CLOSE 1
700 CHAIN"TRAYEAR.BAS"
710 ONERRORGOTO
720 IFERR=77THENRESUME650

0 REM BILLPAY1.BAS
20 REM 1/10/82
30 W =60:CL$=CHR$(22):PRINTCL$
40 PRINT:PRINT"GETTING INFORMATION FOR BILL PAYING
    PROGRAM"
50 OPENOLD"1.YEAR"AS1:GET#1,RECORD1
60 FIELD#1,2ASTN$,2ASTY$,3ASTM$,3ASGL$,40ASN$
70 YZ=CVT$%(TY$):GL$=GL$:M$=TM$:CLOSE1
80 PRINTCL$
90 PRINT:PRINT
100 PRINTTAB(WZ/2-12);" BILL PAYING PROGRAM"
110 PRINT:PRINT
120 PRINT"Enter END to return to MENU"
130 INPUT"Enter two digits for date (01,02)",YI$
132 IF YI$="END" THEN CHAIN "MENU.BAS"
135 IF LEN(YI$)>2 THEN 130
136 IF LEN(YI$)=1 THEN 130
137 IF VAL(YI$)<1 OR VAL(YI$)>31 THEN 130
140 Y$=M$+"/"+YI$
150 READ A,A$,A:IFA$="ENDATA"THEN170
160 X=X+1:GOTO150
170 RESTORE:DIMDZ(X),A$(X),A(X),NZ(X)
180 IFX=0THEN230
190 FORIZ=1TOX
200 READ DZ(IZ),A$(IZ),A(IZ)
210 NEXTIZ
220 PRINTCL$
230 PRINT:PRINT:PRINT
240 PRINTTAB(WZ/8);"Bills will be run for ";Y$
250 PRINTTAB(WZ/8);"Enter END to return to MENU"
260 PRINTTAB(WZ/8);"Enter BEGIN CHECK NUMBER....";
    :INPUTAN$
270 GOSUB820
280 CNZ=VAL(AN$):IFCNZ<1THEN260
290 ONERRORGOTO320
300 OPENOLD"I."+M$AS1:GET#1,RECORD1:FIELD#1,2ASZ$
    :TZ=CVT$%(Z$):CLOSE1
310 ONERRORGOTO:GOTO330
320 IFERR=4THENCLOSE1:OPENNEW"I."+M$AS1:FIELD#1,2ASZ$
    :LSETZ$=CVTZ$(0):PUT#1:CLOSE1:RESUME
330 OPENOLD"1."+M$ AS I
340 PRINTCL$:PRINT"LIST OF PAYMENTS FOR ";Y$:PRINT
350 IFZ=0THENPRINT:PRINT"NONE LISTED AT THIS TIME"
    :GOTO800
360 PRINT"ACCT#    PAYEE                    AMOUNT"
370 PRINT:FORIZ=1TOX
3 0 PRINTUSING"  ###  \12345678901234567890\
    $#,###.##",DZ(IZ),A$(IZ),A(IZ)
390 PRINT:PRINT:PRINT
400 PRINT"Do you want to change $ ";A(IZ);"(Y/N)?";
    :AN$=INCH$(0):PRINT
410 IF AN$="Y" THEN INPUT"Enter new Amount",AX$
420 IF AN$<>"Y" THEN GOTO 470
430 A(IZ)=VAL(AX$):IFA(IZ)<1THEN400
440 PRINT:PRINTUSING'ACCT### \12345678901234567890\
    $##,###.##',DZ(IZ),A$(IZ),A(IZ)
450 PRINT:PRINT"Is this correct? ";:AN$=INCH$(0)
    :PRINT
460 IFAN$<>"Y"THEN400
470 NEXTIZ:PRINT
480 IFI=0THENB10
490 NI=CNZ-1
500 PRINT "SCHEDULE OF PAYMENTS FOR ";Y$
510 PRINT
520 PRINT "ACCT#";TAB(7);
530 PRINT "CHECK";TAB(15);
540 PRINT "PAYEE";TAB(40);
550 PRINT "AMOUNT"
555 PRINT:PRINT
560 FOR IZ=1 TO X
570 IF A(IZ)=0 THEN 640
580 NI=NI+1
590 NZ(IZ)=NI
600 PRINT DZ(IZ);
610 PRINT TAB(7);NZ(IZ);TAB(15);
620 PRINT A$(IZ);TAB(40);
630 PRINT USING '$#####.##',A(IZ)
640 NEXT IZ
650 PRINT:PRINT
660 PRINT "IS THIS CORRECT?";:AN$=INCH$(0)
670 IF AN$<>"Y" THEN RESTORE:GOTO 340
680 FOR IZ=1 TO X
690 IF A(IZ)=0 THEN GOTO 780
700 TZ=TZ+1:RZ=TZ/5:SZ=TZ-(RZ$5)
710 IFSZ=0THENFIELD#1,252ASZ$:LSETZ$=""
    :PUT#1,RECORDRZ+1
720 GET#1,RECORDRZ+1
730 FIELD#1,SZ#50ASZ$,2ASTD$,2ASTC$,2ASTN$,10ASTY$,
    26ASTP$,BASTA$
740 LSETTD$=CVTZ$(DZ(IZ)):LSETTC$=CVTZ$(103)
750 LSETTN$=CVTZ$(NZ(IZ)):LSETTY$=Y$
760 LSETTP$=A$(IZ):LSETTA$=CVTF$(A(IZ))
770 PUT#1,RECORDRZ+1
780 NEXTIZ
790 GET#1,RECORD1:FIELD#1,2ASZ$:LSETZ$=CVTZ$(TZ)
    :PUT#1,RECORD1
800 CLOSE1
810 CHAIN"MENU.BAS"
820 IFAN$="END"THENCHAIN"MENU.BAS"ELSERETURN
830 REM DATA SHOULD BE IN GROUPS OF THREE
840 REM ACCT# DEBIT,ACCOUNT NAME, AMOUNT
850 DATA 201,WARDS,00
860 DATA 204,MASTERCARD CENTER,00
870 DATA 202,SEARS,00
880 DATA 203,FT. WORTH NAT'L. BANK,00
890 DATA 206,VISA CENTER,00
900 DATA 213,WORTHEN BANK-INSTALLMENT DEPT.,00
910 DATA 212,WORTHEN 1ST MORT. CO.,00
920 DATA 205,J. C. PENNEY,00
930 DATA 411,SW BELL CO.,00
940 DATA 405,ARKLA GAS,00
950 DATA 407.AP&L,00
```

```
960 DATA 411,MCI,00
970 DATA 409,LR WATER WORKS,00
980 DATA 106,NORTHERN BANK-IRA ACCOUNTS,00
985 DATA 104,NORTHERN BANK-SAVINGS,00
990 DATA 419,EXXON CORP.,00
991 DATA 425,RIVERSIDE CABLE TV,0
992 DATA 200,DILLARD'S,0
1000 DATA 435,DEMOCRAT,0
1010 DATA 433,ARKANSAS GAZETTE,0
1020 DATA 449,WORLD VISION,0
1030 DATA 449,CHRIST. BROAD. NETWORK,0
1040 DATA 449,CAMP. CR S. FOR CHRIST,0
1055 DATA 401,COLEMAN DAIRY,00
1056 DATA 432,STATE FARM INSURANCE,00
1060 DATA 0,ENDATA,0
1070 END
```

TO BE CONTINUED

DEBUG MOTO D2

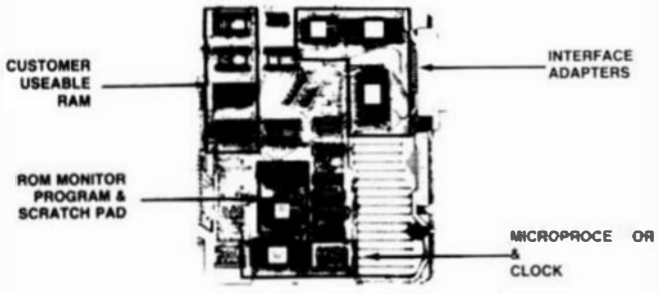
DEBUGGING THE MEK6800D2 MICROCOMPUTER KIT

Lloyd Maul

Motorola Semiconductor Products Sector

The world of microcomputers has been opened to many hobbyists and students with the availability of relatively low cost kits that allow construction of a complete functioning microcomputer system. The MEK6800D2 kit was the original MC6800 based kit put out by Motorola several years ago. This kit has been succeeded by other kits such as the MEK6802D3, MEK6809D4 and MEK6802D5. During the initial life of the D2 kit problem areas were encountered with various customers, generally in the construction phase at that time, and a debugging procedure was developed by the author to assist in resolving the problems. Although the D2 kit is not marketed at the present time the ideas outlined in this article may be helpful for those people that are still using the system and some of the techniques can help people who may be having trouble with the present kits.

The MEK6800D2 is comprised of two printed circuit boards complete with associated parts and an interconnecting ribbon cable. One printed circuit board is the microcomputer system, which includes the microprocessor and its crystal controlled clock, 256 bytes of customer usable RAM, a 1K byte monitor program with associated 128 bytes of scratch pad RAM, two peripheral interface adapters (PIA's) and an asynchronous communications interface adapter (ACIA) as shown in Figure 1A. Additional sockets are included to expand the on board



MICROCOMPUTER MODULE
FIGURE 1A

customer usable RAM to 512 bytes and for insertion of customer defined programs in PROM. The printed circuitry is layed out to accept bus extender devices so that the basic system can be expanded to additional system cards through the bottom connector.

The second board included in the kit (see Figure 1B) is an input/output board for the system. It is capable of keying in information with a hexadecimal keypad and has eight function keys. There are six seven segment displays that indicate the status of the system. This board also provides an audio cassette interface so that programs that have been keyed into memory through the hexadecimal keypad may be stored on and later retrieved from a standard audio cassette. Interface to this board from the microcomputer board is through one of the PIA's and the ACIA. This leaves one PIA available as a user defined program, the I/O board may be disconnected allowing the full I/O capability of the microcomputer board to be available for a given application.

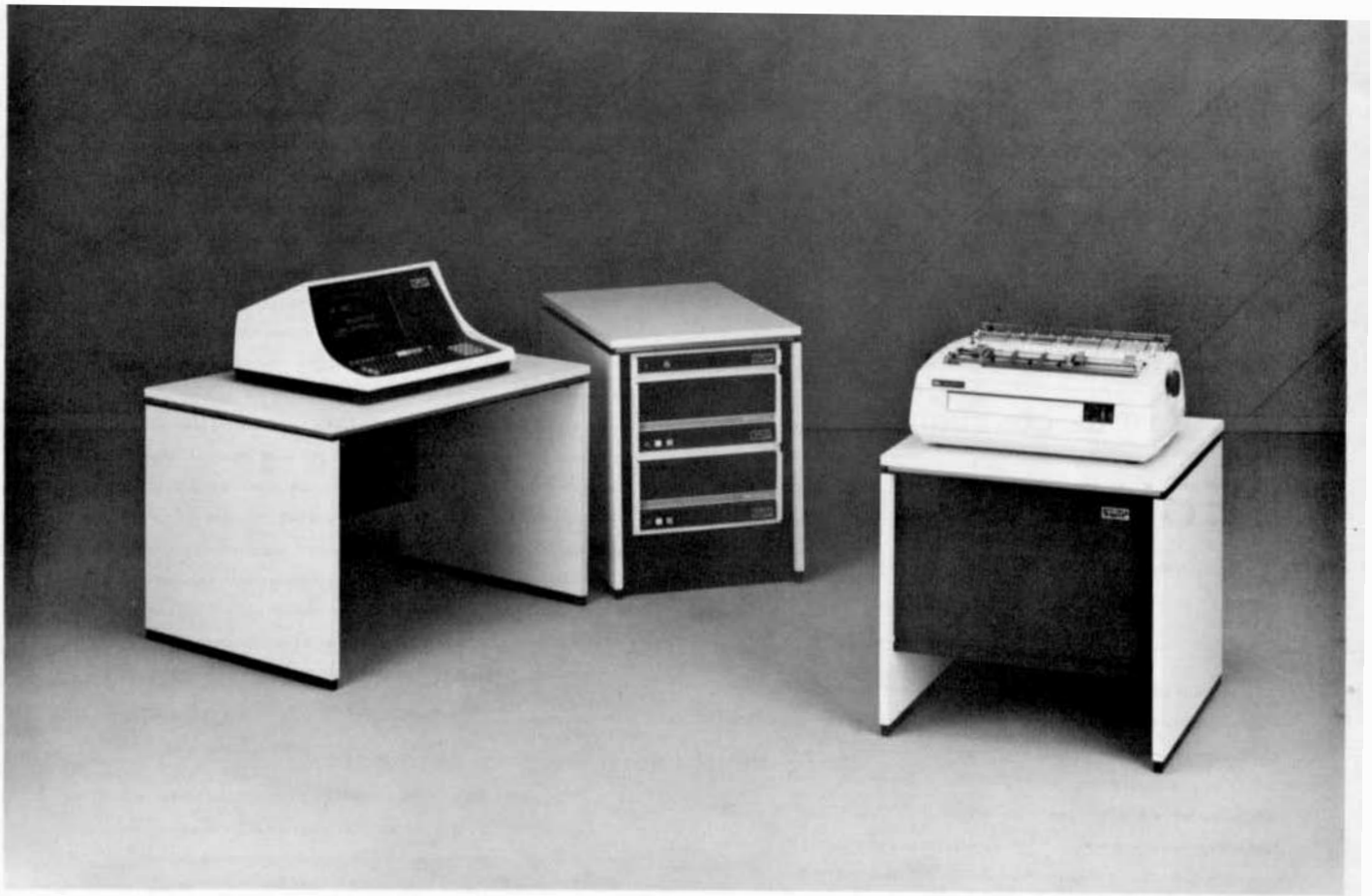
Many kits have been assembled by a variety of people and operated properly the first time power was applied. Obviously such performance gives a great deal of satisfaction and microcomputer functional and programming knowledge can progress immediately from that point. However, things do not always work correctly from the start and some kits have not performed properly with the initial power application. This paper will cover some of the areas that are potential problems and will indicate how debug procedures may be undertaken to isolate various faults. All possible problem areas cannot be covered, but hopefully the techniques discussed will allow the user to find a large number of possible errors. Although this is written with emphasis on the D2 kit, the principles still apply to almost any 6800 system.

Proper Operation

A little extra care with soldering during the construction phase of the kit will normally pay off by saving appreciable time in the debugging phase. Most problems encountered with a malfunctioning system are directly attributable to improper soldering. This has been from application of too much solder, thus bridging across conducting paths as well as insufficient solder and failing to make proper connections where they should be made. At the completion of the construction process, the boards should be given a visual inspection to possibly detect improper solder connections especially noting the key connect points and the cable connections.

After a general inspection and the two boards are properly interconnected by the ribbon cable, power connections may be made through a properly wired connector slipped over the bottom double 43 pin edge connector. Alternatively, power supply wires may be soldered directly to the microcomputer board close to the bottom edge connector fingers or by connecting clip leads to the 100 microfarad filter capacitor also near the bottom edge connector. The I/O board derives its power through the interconnecting ribbon cable from the microcomputer board. The power supply required is a single 5 Volt ± 5% unit. Typical current drain for the two boards is about 1 Ampere. Verify that your power supply is capable of handling this current (preferably greater than 1.5 Amperes). If the power supply has a current limiting capability, insure that it is set above the 1 Ampere level.

Initial turn-on of power will give unpredictable display results even in a properly operating system because of the random nature of device turn-on. A depression of the reset switch should cause the display to go dark if there had been an initial random display. Releasing the reset switch allows the system to execute its start up routine, which initializes the I/O circuitry so that a dash (-) symbol should appear in the left most seven segment display. Display of the dash symbol indicates that the system has properly executed its initializing restart routine and is awaiting a command from the user. At this point it is appropriate to check for proper



THE COMPLETE BUSINESS SYSTEM

+Multiuser+Highly Expandable+Cost Effective

S+ THE CONCEPT

The S+ system is a modular computer system in which all portions of the hardware and software are designed to work together in the most efficient way possible. An S+ single user system with floppy disk storage is a competitive and cost effective entry level system. Unlike most other small computers being sold as "personal", or "small business" machines, the S+ system may be expanded to maximum capabilities using this same hardware and software. You cannot end up with a DEAD END system that cannot be expanded and whose software is not compatible with larger machines. A basic S+ system may be expanded to thirty-two users, a megabyte of main memory and hundreds of megabytes of hard disk storage by simply plugging in, or connecting the desired upgrade equipment.

TOTAL DESIGN—Hardware and Software

The S+ system is an integrated hardware and software design. The two complement and enhance each other in this system. The UniFLEX® operating system used in the S+ systems is patterned after the Bell Laboratories UNIX® operating system, one of the most admired and widely used operating systems in the world. Instead of being an afterthought, the software is part of the design of the S+ system. You can be sure that with this approach that all parts of the computer operate with maximum efficiency and cost effectiveness.

THE CENTRAL PROCESSOR

The basic S+ system is configured with 256K bytes of memory and can be expanded to more than 1 million bytes. An efficient and fast hardware memory management system is used to allocate the available memory among the users on a dynamic basis. As little as 8K bytes, or the entire memory—if needed—can be used by any individual user. This makes it possible to run very large programs on the system, but it also uses no more memory than necessary for a particular job. The increase in cost effectiveness of this system over crude and outdated bank switching arrangements is dramatic.

The central processor runs in both user and supervisor states. It can detect and reject a defective user program. It is impossible for a user program to go bad and stop the entire system, as can happen quite easily in less sophisticated systems.

Task switching is accomplished by use of a multiple map RAM memory, with sixty-four individual task maps. Each task can access from 4 to 64 K-bytes of memory. Multiple tasks may be used in programs that require more than 64K bytes of memory for execution. When a task is completed the memory is automatically released for other use.

SOFTWARE

The S+ operating system, UniFLEX® is a multiuser, multitasking operating system based on the UNIX® operating system that has been used for many years on Digital Equipment Corp. PDP-11 series minicomputers. It is considered one of the most sophisticated and "user friendly" operating systems available. Variations of UNIX® are rapidly becoming standard on mini and larger microcomputers.

A large variety of languages are available for use with the system. These include FORTRAN, COBOL, BASIC, and Pascal. Word processing packages are also available to give you full text processing capability on the system.

Applications programs are available in large quantities in many fields. This includes general business, medical, dental, veterinary, library and real estate management; plus others. Since the system is multiuser it can also be connected to cash registers to produce a point-ofsale terminal system combined with the computer. The possibilities for application of this system are endless.

THE I/O SYSTEM

The S+ system is totally interrupt driven. All terminal and printer I/O devices connect to an I/O bus separate from the main bus. Up to thirty-two separate devices may be connected to the I/O bus at any one time. If I/O activity is great enough to cause an unacceptable slowdown in system operation, a separate I/O processor can be installed in the system. This plug-in option removes all I/O handling overhead from the main processor and allows operation of up to thirty-two external devices at 9,600 baud. Without an integrated total design, as in the S+ system, it would become impractical to use a UNIX®type operating system in a situation with heavy terminal I/O activity.

DISK STORAGE

A wide range of disk storage capacity is available for the S+ system, from 2.5 M-byte floppy disks to an 80 M-byte Winchester and many sizes between. All disk controllers use direct memory access (DMA) type operations to maximize data transfer and to minimize overhead on the main processor. The Winchester disks also use intelligent controllers along with DMA transfers to preserve the performance that these type devices are capable of giving. Without this distributed intelligence the system performance would be greatly degraded. The UniFLEX®operating system is designed to work at maximum efficiency with this type disk system. The data transfer rates achieved by this combination rival those of large minicomputers.

COMMUNICATIONS

A high speed local network communications system is available to interconnect S+ systems. The VIABUS® network will allow communication between systems at data rates of over 400K baud. Such a system makes it possible to share data between local systems in an efficient and low-cost manner.

AVAILABLE SOON

Tape backup—20M-Byte in less than 15 minutes on a standard ¼ inch cartridge.

Mini-Wini—5 and 10 M-Byte Winchesters—5¼ inch package. Winchester performance, for smaller systems in a small package. UniFLEX® compatible design.

Large Capacity—190 and 340 M-Byte Winchesters, plus SMD cartridge drives.

UniFLEX is a registered trademark of Technical Systems Consultants, Inc.

UNIX is a registered trademark of Bell Labs.

VIABUS is a registered trademark of Southwest Technical Products Corporation.

operation of the hexadecimal keypad. This may be done by sequentially depressing the 0 through 3 keys and observing a display of each corresponding key value on each sequential seven segment display. If it does, the E command key may be depressed and should return the user to the dash prompt signal. A repetition of this procedure with keys 4 through 7 would verify the next four keys and so on through the hexadecimal F key. Keep in mind that the hexadecimal B and D values are displayed as small letters b and d respectively. The M command key can be checked by keying in #### (after the dash prompt is displayed) and then depressing the M key. This operation should display the random contents of memory location #### in the last two seven segment displays that follow the keyed in address. Keying in two different hexadecimal values will verify proper operation of the memory change function by displaying the data just keyed in after the second keystroke.

Depression of the G key at this point should advance the address display to ##01 with corresponding data being displayed verifying proper operation. Depress the E command key again and obtain the dash prompt. A depression of the R key should give a full six digit display. This is a stored value in the monitor scratch pad RAM that indicates the stacked value of the program counter and the contents of that memory location if applicable. Depressing the G key now will give a four character display, which represents the index register value. Repetitive operations of the G key will in turn give a two digit display for the A accumulator, a two digit display for the B accumulator, a two digit display for the condition code register, a four digit display for the stack pointer and back to the six digit display associated with the program counter.

Proper achievement of the preceding checks indicates that the main portion of the microcomputer is working as it should. The breakpoint and single step functions may be verified by keying in a short program as indicated in the instruction booklet that comes with the kit. Verification of the audio cassette punch and load operations normally occurs after some familiarization is achieved with the kit and storage of a program is desired on the cassette medium.

TO BE CONTINUED

BIT BUCKET

681 Whitmore Rd., #207
Detroit, MI, 48203
April 24, 1982
(313) 862-3454 (9A-4P EST)

I've been following Ron Anderson's column quite avidly, and have enjoyed his ongoing project of testing various language implementations by calculating the series of prime numbers. Brian Bailey's program in 6809 assembler for calculating primes (published last year) is one of the showcase programs in my system. I recently came across another special kind of number (like primes) that has me intrigued and my experience might interest some of your readers, be they BASIC programmers or assembly language buffs.

I'm taking a class in PL/1 at Wayne State University here in Detroit, and our third program assignment was to write a program that would find the first ten 'perfect numbers.' What's a perfect number? That's the same thing I asked, and I've had more than a little math. It's a number that is also the SUM of its factors. The first perfect number is six (1+2+3=6). The second is twenty-eight. The algorithm is pretty simple, and in two parts. First, determine the factors of the number (those numbers that evenly divide into it), then add them up and if they equal the original number, that's a 'perfect' number. 1+2+4+7+14 = 28. Ten is NOT a perfect number because its factors, 1,2, and 5 don't ADD UP to ten.

Sounds easy, right? So I thought until I started running out of the defaulted maximum cpu time. My program would find the first three numbers (6,28 and 496) in under ten seconds but couldn't find the fourth even with 45 seconds of cpu time. And we're not talking PDP-11 here, we're talking one of the biggies, an Amdahl 470 or an IBM 370. Apparently I wasn't the only student having this problem, so the teacher relented and said if we could find the first three perfect numbers, our algorithm was correct.

Short of burning out my student account a couple of times a day, that was it for trying any further on the school's computer. However, I do have my own computer at home, and though I don't have anything as exotic as a PL/1 compiler on my home system (yet!), I do have BASIC, (and an Assembler program), and the algorithm is really pretty simple, so...

The resulting BASIC program is enclosed, as is a copy of the original program. The original was actually written for a load-and-go version of PL/1, developed at Cornell University, called PL/C. The conversion from PL/C to BASIC was pretty simple, too. It primarily amounted to getting around PL/C's DO WHILE... statement with FOR...NEXTs.

To make a long story short, the BASIC program below has run for TEN HOURS on my 1 MHz 6809 system without finding the fourth number. Unless someone has a radically different algorithm that's much faster, I don't see anyone's 8-bit BASIC coming up with an answer in any reasonable amount of time (TSC's is pretty fast as interpreters go). I suppose I could write a machine language program to do it, but that would be a mammoth project for me (read: I-M-P-O-S-S-I-B-L-E). Until (and IF) someone out there comes up with such a program, does anyone happen to KNOW what the fourth (and fifth, etc.) perfect numbers are? Are they always even (as opposed to odd) as these first three seem to imply?

Keith Alexander

```
)    1   $PL/C ID='KEITH ALEXANDER' ERRORS=(3,3) TIME=(0,25) NOXREF NOSOURCE
)    2
)    3
)    4   PROG3:PROCEDURE OPTIONS(MAIN);
)    5
)    6   /* PROGRAM TO FIND AND OUTPUT A TABLE OF */
)    7   /* THE FIRST THREE PERFECT NUMBERS */
)    8
)    9   /* STRUCTURED VERSION */
)   10
)   11   DCL TEST_NUMB FIXED DEC(4) INIT(2);
)   12   /* NUMBER UNDER TEST */
)   13
)   14   DCL TEST_FACT FIXED DEC(4);
)   15   /* FACTOR UNDER TEST */
)   16
)   17   DCL MAX_FACTOR FIXED DEC(4);
)   18   /* LARGEST VALUE FOR TEST_FACT */
)   19
)   20   DCL QUOTIENT FLOAT DEC,
)   21       SUM FIXED DEC(4),
)   22       TESTCOUNT FIXED BIN INIT(0);
)   23
)   24
)   25   PUT PAGE LIST('TABLE OF FIRST THREE PERFECT NUMBERS');
)   26
)   27   DO WHILE(TESTCOUNT<3);
)   28    SUM=1;
)   29    TEST_FACT=2;
```

```
)   30      MAX_FACTOR=FLOOR(TEST_NUMB/2);
)   31
)   32        DO WHILE(TEST_FACT<=MAX_FACTOR);   /* FACTORING LOOP */
)   33          QUOTIENT=TEST_NUMB/TEST_FACT;
)   34          IF FLOOR(QUOTIENT)=QUOTIENT     /* FOUND A FACTOR */
)   35            THEN DO;
)   36              SUM=SUM+TEST_FACT;
)   37              TEST_FACT=TEST_FACT+1;
)   38            END;
)   39          ELSE TEST_FACT=TEST_FACT+1;
)   40        END;
)   41
)   42      IF SUM=TEST_NUMB
)   43      THEN DO;   /* FOUND PERFECT NUMBER */   '
)   44        PUT SKIP LIST(TEST_NUMB);
)   45        TESTCOUNT=TESTCOUNT+1;
)   46        TEST_NUMB=TEST_NUMB+1;
)   47       END;
)   48
)   49      /* NUMBER NOT PERFECT */
)   50      ELSE TEST_NUMB=TEST_NUMB+1;
)   51
)   52      END;
)   53
)   54
)   55    END PROG3;
@END OF FILE
@RUN &PLC SCARDS=PGM
@EXECUTION BEGINS


TABLE OF FIRST THREE PERFECT NUMBERS
      6
     28
    496

IN STMT   28  PROGRAM RETURNS FROM MAIN PROCEDURE.



IN STMT   28  DYNAMIC FLOW TRACE:
              0019->0011        0013->0018        0019->0011        0013->0018
              0019->0011        0013->0018        0019->0011        0013->0018
              0019->0011        0013->0018        0019->0011        0013->0014
              0019->0011        0011->0020        0020->0021        0027->0007
              0007->0028        0028->0000

 IN STMT   28  SCALARS AND BLOCK-TRACE:

 00000 MAIN PROCEDURE PROG3

 TESTCOUNT=        3     SUM=     496            QUOTIENT= 2.00000E+00
 MAX_FACTOR=     248     TEST_FACT=     249      TEST_NUMB=     497

@EXECUTION TERMINATED
```

```
10 REM
20 REM  PROG3      2-25-82
30 REM
40 REM A PROGRAM TO FIND THE FIRST (T) PERFECT NUMBERS
50 REM
60 REM TRANSLATED FROM A PL/C ORIGINAL
70 REM TO BASIC
80 REM
90 INPUT "NUMBER OF P-NUMBERS TO FIND ",T
100 REM
110 PRINT:PRINT "The first";T;"perfect numbers are: "
120 PRINT
130 N=4:REM NUMBER UNDER TEST--INITIALLY 4
140 REM
150 FOR J=1 TO T
160 REM
170 S=1:REM SUM
180 F=2:REM FACTOR UNDER TEST
190 M=INT(N/2):REM "INT" IS EQUIV. TO "FLOOR" FUNCTION IN PL/C
200 REM M IS THE LARGEST FACTOR TO BE TESTED
210 REM
220 REM FOLLOWING IS THE FACTORING LOOP
230 REM
240 Z=Z+1:IF F>M THEN GOTO 340 ELSE Q=N/F:REM TEST_NUMB/TEST_FACT
250 REM Z IS COUNTER FOR FACTORS TESTED
260 REM Q IS THE QUOTIENT FOR FACTOR UNDER TEST
270 IF Q=INT(Q) THEN GOTO 280 ELSE F=F+1:GOTO 240
280 S=S+F+Q:REM FOUND A FACTOR
290 M=Q-1:REM BUMP MAX.FACTOR DOWN
300 F=F+1:GOTO 240:REM END OF FACTORING LOOP
310 REM
320 REM END OF FACTORING LOOP
330 REM
340 IF S=N THEN GOTO 360 ELSE N=N+1:GOTO 170
350 REM
360 PRINT N:REM FOUND PERFECT NUMBER
370 N=N+1
380 REM
390 NEXT J:REM GO FIND NEXT PERFECT NUMBER
400 REM
410 REM BELL RINGING ROUTINE TO SIGNAL END OF PGM
420 FOR K=1 TO 20:REM TWENTY BEEPS
430 PRINT CHR$(7):REM CTRL-G (BEL)
440 FOR L=1 TO 500:NEXT L:REM APPROX. 0.5 SEC. DELAY BETWEEN BEEPS
450 NEXT K
460 REM
470 PRINT "Total factors tested: ";Z
480 REM
490 END:REM END OF PROGRAM
500 REM
710 REM    The program comes up with the first perfect number
720 REM almost as fast as I can operate a stopwatch, and the
730 REM second within two seconds. The third number, 496, isn't
740 REM on the screen until a little over five minutes after
750 REM the program started. Small computers being cheap to run,
760 REM I put the beeping routine at the end and let the program
770 REM run OVERNIGHT. When I awoke TEN HOURS later, it still
780 REM hadn't come up with the fourth number.
790 REM
800 REM   I stopped the program, and asked for the current values
810 REM of the program variables. The number under test was
820 REM 7109, the factor under test was 2822, and the maximum
830 REM factor to test was shown correctly to be 3554.
```

First (Organizational) Meeting of Users of OS-9

May 16, 1982. The organizational meeting of the user's group began on an informal basis, with Brian Capouch presiding. Brian solicited suggestions from the group concerning the form and substance of his proposed user society. Several of the attendees offerred suggestions ranging from heavy subsidies required of manufacturers to commercial operation of the group. The general consensus was that the group should stand on its own without being supported by either Microware or Motorola. The staff of Microware expressed tremendous relief.

At this point in the meeting, there was a call for election of provisional officers to organize the formal user's group. Four officers were nominated and elected by acclaimation. The officers then gave their addresses and phone numbers, as follows:

President: Brian Capouch, Rural Route 1, Box 270, Monon, Indiana 47959. (219) 253-8181

Vice President: Sheldon Epstein, Epstein Associates, PO Box 400, Wilmette, Illinois 60091. (312) 564-9292

Secretary: Howard Harkness, Word's Worth, PO Box 28954, Dallas, Texas 75228. (214) 321-9285

Treasurer: Terry Straehley, Straehley Associates, 1005 Roble Lane, Santa Barbara, California 93103. (805) 962-6701

Volunteers for the steering committee are: Steve Bush (703) 237-2000 ext 7121; Mike Pagett (305) 678-6776; Tom Westhoff (612) 235-5971 (wk) or 235-6096 (hm); Tom Henson (219) 258-1000; Jean Merrill (707) 994-3019.

Volunteers for the Communications Committee are: Terry Ritter (512) 837-2494; Mark Kroll (800) 328-3336 (Intercomp); Sam Epstein (312) 564-9292; Rich Andrewski (405) 721-2699; Ray Cadmus (816) 263-6693 (wk) or 263-1228 (hm).

Any person wishing to make suggestions to the steering committee may write Sheldon Epstein, PO Box 400, Wilmette, Illinois 60091. Of course, preference will be given to those suggestions made by paid members.

After the meeting, the officers were called into an executive session by the president. It was resolved to make future meetings of the officers via telephone, with emphasis on quick turnaround. The vice president was assigned the task of taking possession of any hardware donated at the seminar. Action for incorporation was deferred to the steering committee, so for the moment, we remain a voluntary association.

2403 Perry Lane
Alvin, Tx 77511
May 1, 1982

Dear Dan:

I guess it's time for my annual attempt to contribute something useful. As always, feel free to use as little or as much of it as you think appropriate. This time it's a simple hardware modification for those of us who're still somewhat overwhelmed trying to use up all the capabilities of the old SWTP 6800. The modification (explained more thoroughly in the accompanying article) enables one to use the memory space from 8020 to 9FFF for RAM memory. Everyone will probably have a different use for this space! I like to run the SWTP Dis-Assembler there so I'll be sure it doesn't conflict with any programs I'm working on.

Keep up the good work, and let's here from some more 6800 users, especially those of us using the JPC cassette system!

Sincerely yours,

Charlie Hollpeir
(713) 585-2265

ADD 8K OF RAM TO YOUR SWTP 6800

Why add another 8k to the SWTP 6800? Just to see if I could do it, and because of the certainty that Software, like Bureaucracy, will expand to fill all available space. Actually, I had my SWTP 6800 system pretty well loaded, with 8k of EPROM and 48k of RAM, but that 8k wasted by the I/O ports continued to bother me. What this article describes is a fairly simple way to fully decode the I/O addresses and use a spare 8k board to provide RAM from 8020-9FFF, giving 8k (less 32 bytes) of usable memory.

The whole thing starts with something mentioned by Pete Stark a few years ago, in that other magazine formerly known as Kilobaud. To fully decode the motherboard, you need only 2 chips, a 7425 (Dual 4-input NOR) and a 7400 (Quad 2-input NAND). You only need one gate from the 7400 for this part, but two more gates will be used later. The schematic hook-up is shown in Fig. 1.

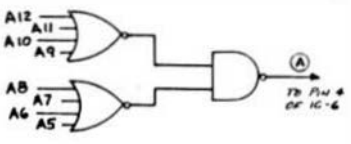


Figure 1. Motherboard Modifications for Full I/O Decoding

To explain the circuitry a little, the SWTP decoding scheme for the motherboard uses line A5, in combination with lines A15, A14, and A13 to decode (enable) I/O ports whenever an address between 8000 and 9FFF appears on the address bus. This decoding (or rather, partial decoding!) is handled by the 74LS138 chip IC-6. Line A5 is tied to the xCS (read not CS) pin of IC-6, so that the I/O ports on the first motherboard of a two-motherboard system are accessed only for line A5 low (0). If two motherboards are used, the second board is set up to be accessed for A5 high, by grounding the xCS pin and tieing A5 to a CS pin instead.

The 74LS138 decoding logic works by looking at the three lines tied to the input pins (in this case A15, A14, & A13), and decoding them as a binary number. So for A15 high and the other lines low (binary 100x), the decoded number is decimal 8, which is also hexadecimal 8, or corresponds to the hexadecimal addresses of 80xxx. It just so happens that hex 9 is a binary 1001, so addresses from 8000-9FFF are all used by the original 32 (or 64 if using two motherboards) memory locations for I/O. See Table 1.

Hex Address	Binary value of address A15		A5	A0
	v		v	v
8000	1000	0000	0000	0000
800F	1000	0000	0000	1111
801F	1000	0000	0001	1111
8020	1000	0000	0010	0000
802F	1000	0000	0010	1111
803F	1000	0000	0011	1111

Table 1. I/O Addresses

Since I don't have a second motherboard, and can see no possible future need for one, all the memory space above 801F should be available for RAM. Pete's circuit, Fig. 1, will handle the motherboard. All that's necessary is to wire it up and connect from point 'A' to pin 4 of IC-6, and also to cut the motherboard trace going to pin 6 so we don't get conflicting chip enabling signals. The easiest way to do this (if IC-6 is soldered in, as mine is) is to cut the pin just above the board, on the top side of the board, and carefully bend up the pin stub so that the jumper wire can be soldered to it.

But that's only half the story. What I started out to do was to get RAM from 8020 to 9FFF. To do that, I need the signal generated at point 'A'. The reasoning is, if this signal enables only the 20 bytes dedicated to I/O, then it could also be used to disable those same 20 bytes on a RAM board. Because of this, I actually placed the circuits on the 8k SWTP MP-8M board (and because the address lines were readily available on the board, and hard to get to on the motherboard).

The MP-8M board decoding logic is similar to that of the motherboard in that a 74LS138 chip is used to select which 8k boundry of memory is being accessed. (See Fig. 2.) The MP-8M can be made to reside at any 8k boundry, but the on-board switches provide for only the lower four 8k sections. We need to use pin #11, which has no connections on the board in the original configuration, to enable the board for the 8k boundry of 8000-9FFF. This can be done simply by jumpering from Pin #11 of the 74LS138 (chip 1B), to the common trace on the DIP switch. However, doing only this causes conflict between memory addressed at 8000-801F and the I/O ports. What we need to do is add additional logic between pin #11 and the DIP switch common connection trace to enable the RAM only when I/O is not being addressed. This information is available at point 'A', Fig. 1. Since pin 11 of IC-1B goes low to enable the 8k board RAM, and since point 'A' must be high whenever RAM is to be accessed, a truth table such as Table 2 can be constructed. This table leads to the circuit shown in Fig. 3. Since the board is enabled for pin 11 low, the additional circuits must provide a low signal to the board for addresses 8020-9FFF, but a high signal for 8000-801F.

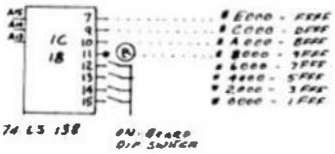


Figure 2. MP-8M Decoding

A	B	C
0	0	1
0	1	1
1	1	1
1	0	0

Table 2. Truth Table

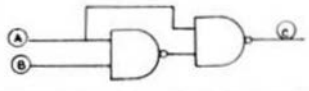


Figure 3. Additional Logic Required to Avoid I/O Conflict

The easiest way to handle the wiring is to do it right on the back side of the board. Epoxy the two chips upside down (pins pointing away from the board), and wire directly to the pins. A wire wrap tool can be used on the pins, if handled carefully. I added a dab of solder for insurance after everything checked out.

Just be sure to label each chip before you glue them down, and be careful when counting pins, since they're upside down. Fig. 4 shows the pin numbers used. I used a jumper wire with a molex pin connector for the connection from point 'A' to pin 4 of IC-6 on the motherboard, so the 8k board is fully reusable. All that's required is to disconnect the jumper wire, remove the 8k board, and place the jumper on the A5 pin on the motherboard, and I'm back in business with a standard SWTP 6800 system, less 8k of memory.

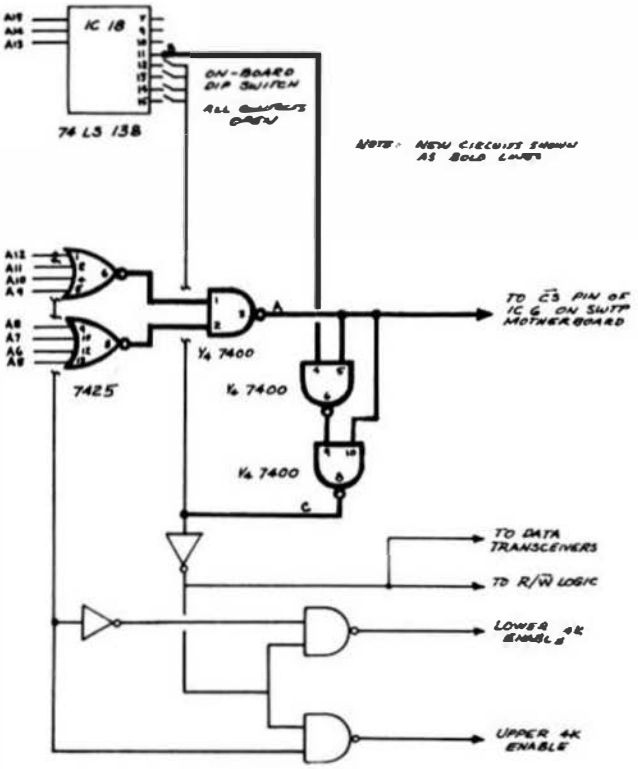


figure 4.
Decoding Logic for Modified MP-8K Memory Board

The Boolean logic for the modifications follows:

For Fig. 1:

$$\overline{A12+A11+A10+A9} \cdot \overline{A8+A7+A6+A5} = A$$

or $A12+A11+A10+A9+A8+A7+A6+A5 = A$

Which means that A is high if all of the lines A12 thru A5 are high. (Or, that A is only low when any of the inputs are low. Now as we saw earlier, the 74LS138 chip IC-6 will give an enabling signal (low) for the I/O only if the binary address 100x is on the input pins, and the $\overline{CS}$ signal (A) is low.

The Boolean logic looks like this for the gates shown in Fig. 3.

$$\overline{\overline{A \cdot B} \cdot A} = C$$

or: $\overline{\overline{A \cdot B} + \overline{A}} = C$

or: $(A \cdot B) + \overline{A} = C$

Which of course gives the Truth Table shown above.

It should be apparent that this scheme could work just as well, with only slight modifications, if one wanted to relocate the motherboard I/O to some other memory locations. For example, since only 920 locations are now required for I/O a logical place would seem to be the first 920 bytes of page one, $0000-001F. This would give Direct Page addressing for all I/O, which would seem to me to be a logical choice, if not for the incompatibility with all of the existing 6800 hardware. What a shame someone at SWTP didn't dedicate those 32 bytes for I/O! Of course any other relocation runs into the same incompatibility problem, but for someone with a special system, or say, someone converting a SWTP 6800 system to the 6809, this might be a useful modification. For example, if for some reason you wished to relocate I/O to the first 920 bytes of the 4k section of $9000-$9FFF, all that would be needed would be to invert the A12 line prior to the 4-input NOR gate. This change would turn I/O on only for addresses $9000-$901F, and would turn the 8k RAM off only for those same locations. To switch to a different 8k section, such as $E000-$FFFF would require changing additional wiring on the motherboard, but wouldn't be too difficult. In this case, you'd have to cut the trace from pin 36 of IC-8, and jumper this trace to pin 37 (refer to the memory locations shown in Fig. 2. The same 8k boundries apply for the motherboard).

MIDDLESEX POLYTECHNIC
PSYCHOLOGY LABORATORIES
QUEENSWAY, ENFIELD
MIDDLESEX, EN3 4SF
ENGLAND

Dear Mr Williams,
Could you please publish this omission to my article on page 35 of April's issue.

The following should be inserted at the end of the chapter on background:-

As we are using SWT CT 82's we have converted the SWT MP-S serial cards to run at the high baud rate operation described in appendix L of the CT 82 manual.

Sorry about this. I hope that nobody has been caused any problems by this omission.
While I am on the subject of CT 82's, early models imported into the UK were fitted with a 60 Hz ROM (CT 82-A1), the UK having a 50 Hz electricity supply. This caused the screen display to flicker. SWT have produced a 50 Hz ROM (CT 82-B1) to correct this problem. I hope this last piece of information is of use to those CT 82 users in 50 Hz areas.
Yours Sincerely,

Brian Roberts

BRIAN ROBERTS

buffers which are used when the card is driving the SS-50 BUSS. In this mode of operation the power requirement is superseded by the importance of drive power for the SS-50 BUSS. The card supports the full sixteen bits of address plus three extended address lines. Since the 146805 only supports 8K, the memory map of the SS-50 BUSS is broken into 2K blocks and paged by the 6805 card. The card comes with a timer, two programmable 8-bit ports, three optional 8-bit input ports, four optional 8-bit output ports, a 2K EPROM monitor, a 2K STATIC CMOS RAM, a DB-25 RS-232 PORT (5V), and a wall plug type power supply. For more details see our ad in this magazine.

THE MICRO WORKS

NEW PRODUCT RELEASE

The Micro Works is pleased to announce the release of Color Forth, a high-level language for the Radio Shack Color Computer.

Color Forth is a highly interactive language like Basic, with the structure of Pascal and execution speed close to that of Assembly Language. You will find many advantages in using Color Forth: it will be faster to write programs than using Basic or Assembly, and execution speed is 5 to 20 times that of Basic!

Color Forth consists of the standard FORTH Interest Group (FIG) implementation of the language plus most of FORTH-79, and a super screen editor with split screen display. Mass storage is via cassette interface. Color Forth also contains a decompiler and other aids for learning the inner workings of this fascinating language. It auto-configures to 4K, 16K or 32K models. Color Forth contains 10K of ROM, leaving your RAM for your programs! The 112-page manual describes the many hardware-specific features of the implementation, including a glossary of the system-specific words, a full standard FIG glossary and complete source listing. Color Forth, written by Talbot Microsystems, comes to you in ROMPACK for $109.95.

P.O. BOX 1111 DEL MAR, CA 92014 714-942-2400

technical systems consultants, inc.

6 May 1982

Don Williams
c/o 68 Micro Journal
PO Box 849
Mixson, TN 37343

Dear Don,

One of the few problems we have had reported in UniFLEX over the last year was concerning the "printer spooler." We had several reports from users of problems, but no one could give us a procedure in which we could duplicate the problem. We spent many hours investigating this, but we could never force the trouble in the lab. At the recent SWTPc distributors' meeting, one user finally gave us a sequence which would produce "garbage" output. This sequence, in actuality, causes two printer spoolers to be activated. Both spoolers print the same files, at the same time, which definitely causes "garbage" output.

The solution to this problem is to always wait 30 seconds after using "pstop" before reactivating the spooler. The next version of UniFLEX will do this waiting for the user.

We are sorry this problem has existed. It is, however, next to impossible for us to find a problem unless we can see the problem.

Sincerely,

Dave Shirk

DS/jg

111 Providence Road • Chapel Hill, North Carolina 27514 • (919) 493-1451

META LAB COMPUTERS • PERIPHERALS • SYSTEMS

2888 Bluff Street Suite 106 • Box 1559 • Boulder, Colorado 80306 • (303) 499-4236

OS9---FLEX COPY UTILITY

METACOPY 130 is a file copy utility program which runs as a task under the OS9 operating system. The program is menu driven for ease of use. The utility permits the user to communicate with the shell of OS9 at any time. The program enables a user to transfer any type of OS9 file to a FLEX sequential file, and to transfer any type of FLEX sequential file to an OS9 file. It also provides the ability to copy OS9 files to OS9 files on the same or different disks, and FLEX files to FLEX files. The utility also provides for a directory of FLEX disks while in the OS9 operating system. This program will be especially useful for copying program source from one operating system to another. Price $135. Delivery from stock.

CP/M FOR SS50 BUS USERS

Meta Lab announces the Z809 SOFTBOARD SYSTEM for 6809 based Southwest Technical Products Corp., Gimix Inc., and Smoke Signal Broadcasting computers. By adding the Z809 microprocessor board and CP/M 2.2 to these systems, a user can immediately run over 2000 CP/M application programs. Programs like Wordstar, dBase II, Supercalc, and Peachtree Accounting Software. The Z809 will greatly expand the SS50 Bus users library of software programs.

The Z809 is easy to install. Plug the processor board in an unused slot, boot the system with the disk, and you are up and running CP/M. No reconfigurations are required for standard SS50 systems. The Softboard System is compatible with standard CP/M formatted disks so that software is easily exchanged or ported to other systems.

Each Z809 Softboard System comes with a 50 pin processor board, CP/M 2.2, Editor, Assembler, Debugger, and complete reference manuals. Total system price - $595.

The Z809 Softboard System will be demonstrated at the National Computer Convention, June 7 - 10, at booth A101.

CP/M is a trademark of Digital Research Corporation.

Art Weller
3217 Paseos Court
El Paso, TX 79904
(915) 755-2516

April 23, 1982

68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Mixson, TN 37343

Dear Don,

Here's a fix for the "swap-the-disks" problem in Flex. Pete Stark described the problem and its cause with great clarity in the April issue so I'll not repeat it here.

Randy Kron, of Kalona, Iowa, came up with the solution. I have merely converted to a more universally useable form which Randy suggested I send in. The enclosed listing is somewhat unorthox, but I figured it would save some column-inches if you decide to print it.

The program first picks up the drive number from a CLOSEd (or DELETEd) file. Loops through all the open FCB's to see if any files are still open on that same drive and if so, is it for WRITE? If any WRITEs are still open, never mind. If all are closed, null out the first two bytes in the drive parameter table. This will force an update from disk for the next WRITE.

A still better method would be to re-assemble the FMS as this would allow use of a lot more of the existing code, reducing the added code to a relatively few bytes. If someone works up the ambition to do this, here is a suggestion. Consider adding four more bytes to the table (one for each drive) and increment/decrement as WRITE files are opened/closed. Test this byte to see if the table should be updated from disk. If anyone tries this, I'd be interested to hear from them.

Yours truly,

Art Weller

A. Weller

```
                        TTL    FIXSIR
                    *
                    * BY RALPH KRON, APR 6, 1982
                    * "UNIVERSAL" VERSION BY ART WELLER, APR 22, 1982
                    *
                    * REMOVE ALL BUT ONE SET OF EQUATES
                    * BY CHANGING THE OTHERS TO COMMENTS.
                    *
                    * ADDRESSES HAVE BEEN TESTED BUT MAY
                    * BE DIFFERENT IN OTHER VERSIONS.
                    * RE-CHECK THEM!!!
                    *
                    * NOTE:  DRVX6 IS AN EXISTING SUBROUTINE,
                    *        THAT CALCULATES A POINTER TO
                    *        THE DRIVE TRACK/SECTOR PARAMETER
                    *        TABLE.
                    *
                    * MINIFLEX EQUATES
                    *
                    * CLSFIL EQU $7D68 CLOSE
                    * DELFIL EQU $7E3D DELETE
                    * DRVX6 EQU $7B1B SEE NOTE
                    * FCBBAS EQU $7809 FCB BASE POINTER
                    * CURFCB EQU $780B CURRENT FCB
                    * FLXEND EQU $7EA2 END OF FMS
                    * FMSTBL EQU $789D FMS FUNCTION ADDRESS TABLE
                    *
                    * FLEX2 EQUATES
                    *
BAE8                CLSFIL  EQU     $BAE8
BBEE                DELFIL  EQU     $BBEE
B79C                DRVX6   EQU     $B79C       SEE NOTE
B409                FCBBAS  EQU     $B409
B40B                CURFCB  EQU     $B40B
BFC0                FLXEND  EQU     $BFC0
B4D4                FMSTBL  EQU     $B4D4
                    *
                    * FLEX9 EQUATES
                    *
                    * CLSFIL EQU $DAC8
                    * DELFIL EQU $DB0B
                    * DRVX6 EQU $D788 SEE NOTE
                    * FCBBAS EQU $D409
                    * CURFCB EQU $D408
                    * FLXEND EQU $DFBD
                    * FMSTBL EQU $D4CE
                    *
                    * FCB OFFSETS
0002                ACT     EQU     2           ACTIVITY STATUS
0003                DRV     EQU     3           DRIVE NUMBER
001C                FLP     EQU     $1C         FCB LINK POINTER
                    *
                    * THE FOLLOWING MODIFIES THE FMS
```



```
                    * TABLE OF FUNCTIONS.
                    *
B4DA                        ORG     FMSTBL+$6 FORTH ENTRY (CLSFIL VECTOR)
B4DA BF C1                  FDB     NEWCLS
B4EA                        ORG     FMSTBL+$16 12TH ENTRY (DELFIL VECTOR)
B4EA BF C6                  FDB     NEW DEL
                    *
                    *
BFC1                        ORG     FLXEND+1    FIND A HOLE TO PUT THIS IN
BFC1 BD BA E8       NEWCLS  JSR     CLSFIL      USE ORIGINAL CODE AS SUBRTN.
BFC4 20 03                  BRA     ERASE       NOW FIX SIR
                    *
BFC6 BD BB EE       NEWDEL  JSR     DELFIL      USE ORIGINAL CODE AS SUBRTN.
BFC9 07             ERASE   TPA                 SAVE ERROR STATUS
BFCA 36                     PSH A               ..IN CC
BFCB 37                     PSH B               ..AND IN B
BFCC FE B4 0B               LDX     CURFCB      GET NUMBER OF DRIVE
BFCF A6 03                  LDA A   DRV,X       ..JUST USED
BFD1 FE B4 09               LDX     FCBBAS      START OF FCB CHAIN
BFD4 27 10          ERASLP  BEQ     DOERAS      END OF CHAIN, DO FIX
BFD6 8D 19                  BSR     FNDFCB
BFD8 A1 03                  CMP A   DRV,X
BFDA 26 06                  BNE     NXTERS      WRONG DRIVE
BFDC E6 02                  LDA B   ACT,X
BFDE C1 02                  CMP B   #2          OPEN FOR WRITE?
BFE0 27 0B                  BEQ     NOERAS      YES, FORGET FIX
BFE2 EE 1C          NXTERS  LDX     FLP,X       GET NEXT FCB
BFE4 20 EE                  BRA     ERASLP      ..AND LOOP
BFE6 8D B7 9C       DOERAS  JSR     DRVX6       GET ADDRESS OF SIR INFO
BFE9 6F 00                  CLR     0,X         CLEAR THEM
BFEB 6F 01                  CLR     1,X         ..BOTH
BFED 33             NOERAS  PUL B
BFEE 32                     PUL A               RESTORE CC
BFEF 06                     TAP
BFF0 39                     RTS
                    *
                    * THE FOLLOWING FINDS THE START OF
                    * A FCB WHEN X IS POINTING TO THE
                    * LINK BYTES. IT IS REDUNDANT SINCE
                    * THIS CODE IS ALREADY IN FLEX, BUT
                    * NOT IN THE FORM OF A SUBROUTINE
                    * (UNLESS IT IS REASSEMBLED)
                    *
BFF1 C6 1C          FNDFCB  LDA B   #FLP        NUMBER TO BACK UP
BFF3 09             FNDFLP  DEX
BFF4 5A                     DEC B
BFF5 26 FC                  BNE     FNDFLP
BFF7 39                     RTS
                    *
                    * THE FOLLOWING, MORE COMPACT, 6809 CODE
                    * MAY BE SUBSTITUTED FOR ABOVE.
                    *
                    * ERASE PSHS CC,B ACTIVITY STATUS
                    *   LDX CURFCB GET NUMBER OF LAST
                    *   LDB DRV,X ..DRIVE USED
                    *   LDX FCBBAS START OF FCB CHAIN
                    * ERASLP BEQ DOERAS END OF CHAIN, DO FIX
                    *   CMPB DRV-FLP,X
                    *   BNE NXTERS WRONG DRIVE
                    *   LDA ACT-FLP,X
                    *   CMPA #2 IS FILE OPEN FOR WRITE?
                    *   BHS NOERAS YES, DON'T FIX SIR
                    * NXTERS LDX 0,X GET NEXT FCB
                    *   BRA ERASLP ..AND LOOP
                    * DOERAS JSR DRVX6 POINT TO SIR WORK AREA
                    *   CLR 0,X CLEAR FIRST AND
                    *   CLR 1,X SECOND BYTES
                    * NOERAS PULS CC,B,PC RESTORE STATUS AND RETURN
                    *
0037                BYTES   EQU     *-NEWCLS
                            END

NO ERROR(S) DETECTED
```

9607 EXORbus* Table Top Enclosure

The 9607 is our newest Table Top Enclosure for office use with EXORbus microcomputer cards. The 9607 provides 16 slots and a system power supply in an attractive 20"x 8½"x20" package. It comes in ready-to-use form complete with fan and circuit breaker and the rear panel is prepunched to accept a variety of I/O connectors. Best of all it's a twin package to our line of 10, 20, and 40 megabyte Winchester Disc Storage module. The single quantity price for the 9607 is $995.00.* Delivery is from Creative Micro Systems, 3822 Cerritos Avenue, Los Alamitos, California 90720. Telephone (213) 493-2484 or (714) 761-2995.

Dear Editor,

I have just completed the installation of eight 64k ram chips into the TRS Color Computer. The modification was to an early version of the circuit board. None of the 32k jumper points were available. I was not as lucky as Frank Hogg reported in one of your recent journals.

The modification is not simple, but with a little care and patience the work is fairly easy. About half way through you will wish that you had dropped the cash to let Radio Shack perform the modification. Unlike Frank Hogg, who had all the work done for $99.00 and got a working 64k computer, I contacted the local Radio Shack. They said the installation would be $149.00 plus $50.00 for installation. This would only guarantee the page 0 32k of ram. With the possibility of having to put out another $102.00 for the ram I opted to take on the conversion myself. I do not include the following as a complete step by step procedure, but more as a " I got more than I bargained for" procedure to help make the decision. The total for my modification was $102.00 for the first quality TMS 4164-25 two hundred fifty nanosecond access time parts. They seem to work well.

The details of the modification are as follows: 1) remove the outside cover, 2) remove the three screws which hold down the power supply, 3) disconnect the wires which attach the power supply unit to the main circuit card,4) pull the power supply, 5) disconnect the keyboard plug 6) pull the keyboard from the case, 8) remove the circuit board from the case.

You are almost there. The bottom of the circuit board is held away from the board by several nylon standoff mounts. Carefully note the location of all (about 16) push in fasteners. It might be a good idea to use a small amount of silicon sealant to connect the standoffs to the circuit board. Make sure that the sealant is not conductive.

Remove the push in shield fasteners that hold the shield to the bottom of the circuit board. At this time, you should have access to the bottom traces of the printed circuit card. You will also have a feeling --why didn't I just spend the money and have it done. You are at the point where you can identify the traces which must be cut. The 32k boards already have the etch cut and they have jumpers that all are available from the top. If you have a board which has 32k jumpers marks, then you just wasted a lot of time. You must locate three bus wires. One of the busses is used to connect to pin 1 of the RAM circuits. This bus is unused with some types of 64k chips. The bus is grounded with some chips and floating with others. In order to not take any chances, I just cut the buss slightly to the power supply side of the last RAM chip. The buss was then grounded. The ground buss is immediately below the pin 1 buss. The second buss line connects all pin 8's of the RAM together. This bus is cut just to the power supply side of the last RAM chip. This line is then connected to the +5 volt buss line. The third line to be cut was the +12 volt supply which went to pin 9's of the RAM. This is also cut to the power supply side of the last RAM. Make sure that you include the RAM chips, but not any of the pull up resistors for the chip which is beside the 8 RAM circuits. The pin 9 buss is connected to the 74LS783 synchronous address multiplexer pin 35. A 33 ohm resistor is in series with the address line. Eight bypass capacitors c31,c35,c48,c45,c61,c64,c67 and c70 must be removed. On the 32k ram, pin8 is +5vVcc, pin 9 is address bit 7, and pin 1 is n/c. A resistor is connected in series with the address line to eliminate noise. The jumper that is not near the PIA chips is connected for the 16K side of the strap. The second jumper is labelled 4 or 16. The jumper selects TTSL. This is not listed in the green Radio Shack "Color Computer Reference Manual", but a call to Radio Shack identified the wire as going to pin 17 of U8. A jumper goes from the center pin of the three pins together to pin 17 of U8. The old jumper is removed. At this point you have completed the main changes. Be sure to check for an open foil where you made your cut. Be sure that there are no chips floating around. Carefully inspect you work for solder bridges and other problems.

Replace the insulator over the bottom of the circuit board. Replace the metal shield. Be sure that you don't have an inadvertent short. Aeplace all the push in fasteners. Check the power supplies for a direct short circuit. The shield is multiply grounded and once you have more than one screw in place, you can't check for continuity any more.

Replace the circuit board into the case. The screws are replaced. Replace the power cord. Note that one of the cords was marked red. On mine, the red cord went to the outside corner, the green to the center, and the other hot line to the third pin. Replace the other wire jumpers from the pc card to the power unit.

Replace the keyboard and plug it into the main circuit board. The type of wire used is very fragile, don't even look cross eyed at it or one of the wires will break.

Carefully check everything. You are now ready to replace the Version 1.0 rom with the version 1.1 rom. If your machine works with the games paddles on the CHESS game you probably have the version 1.0 rom. I guess Radio Shack screwed up. The game paddles don't work right with the 1.1 rom. The Version 1.1 rom cost $24.31 at Radio Shack.

You can now add the simple revisions as described by Frank Hogg in the MaY issue of 6809 Microjournal. Once the revisions are in, and you have everything correct your machine will be up and running with 64k ram.

The programs supplied by Frank Hogg can be used to check the RAM. It checks everything up to FFFF. The second program, available in Color Computer News, February 1982 gave a move rom routine. The program will allow you to run, check, and modify RAM BASIC. When these play you are there.

Don't rely on this paper for all of your information. Be well prepared with the "Color Computer Reference Manual", Frank Hogg's paper on the subject of conversion, and any other information available. Be careful where you cut the etch.

I don't wish to appear too negative, but the task was more than I had bargained for at the start. I assumed that all the boards were the same, but even though my machine is less than 3 months old, the old version rom, and a low serial number was involved. None of the under board work is required on the newer boards. My machine worked the second time I tried it. The first time, I forgot to plug the power cord into the wall. It sure gives one a good feeling when the job is done. The total down time of the machine was approximately four hours.

Yours Truly,

Donald J. Sommer

Donald J. Sommer
3931 South Burns Street
Seattle, Washington 98118

Salt Lake City, Utah

May 14, 1982

Dear Mr. May:

I've run across a bug in Radio Shack's Color Computer BASIC that could have frustrating, if not serious, results if one is not aware of it.

The problem is as follows:

If you have two numbers, A and B, whose ratio, A/B, is equal to or greater than 2^{128}, then you get the following results:

$$A + B > B$$

$$A - B = -B$$

Evidently, when the difference in exponents in memory is greater than or equal to $80, then the larger number is evidently treated as if it were zero.

The problem does not immediately occur when the smaller number is ordered first, as in

$$B + A \text{ or } -B + A .$$

This seems to correct the problem, although I have run a program, making this correction, and the problem still showed up, although much later in the program.

I called Radio Shack concerning this problem and got the reply that they were aware of it, but had no suggestions.

In all fairness, other than this bug, I have enjoyed using the Color Computer and I consider it, for the price, to be one of the best systems currently on the market.

I have enjoyed your column and have found the hardware descriptions to be very informative.

Sincerely,

David G. McDonald

David G. McDonald
Meteorology Department
University of Utah
Salt Lake City, Utah
84112

Dear Mr. Williams,

Somehow a line of code was lost in my printout of the "BIMOVE" utility published in the June 82 issue of your magazine. Without this line, the code will not run.

This line should be added:

A19E D4 FCB 4

William H. Hall

POSITION WANTED

Working manager or programmer/analyst position with a dynamic software development team. Prefer mini/micro systems but would consider other interesting projects.

BACKGROUND: 20+ years varied computer experience much of it large scale IBM systems. Last several years have concentrated on interactive DEC systems, small IBM systems and microcomputers.

OPERATING SYSTEMS: IBM - OS/DOS, DEC - TOPS20. Micro - OS9, FLEX, CP/M, TRSDOS, APPLE-DOS.

LANGUAGES: COBOL, BASIC, RPG, Pascal, C, Forth (a little), and several assemblers. (IBM - BAL, DEC - Macro20, 6809, 6502, 8080)

APPLICATIONS: Data communications, Inventory control, Retail operations, POS data collection, general financial, hospital accounting, municipal utility billing, county tax collection etc.

CONTACT: Ray Cadmus (OS9 NOTES)
600 W. Lee
Moberly, Mo. 65270
(816) 263-6693 / 263-1228

CLASSIFIED

** CLASSIFIED **

6800 CPU Card $100.00. AC-30 Cassette interface with documentation $50.00 DC-3 Controller Card $100.00
LEW 1-615-842-4601

*****

Tano 56K Dual Serial Ports, Dual 5" Disk Drives, complete software package. Like New; Wholesale $5300 - will take $2995. Three SWTPC 8K Memory Boards $95 ea. PR-40 Printer, Extra Paper, and Ribbons; $95.
LEW 1-615-842-4601

*****

SWPTC 6800, 24K, SSB-BFD-3, 24X80, much more. Beautiful System only $1100. Call for details.
Rick Hadley Day(319)475-2201 Evening(319)472-2278

*****

Smoke Signal 6809 system w/OS9-1, Basic09, Assembler, Editor, Debug, 62KRAM, DCB4A, 2Ser/2Par., 2DS/DD MPI52DD, less than 1 year old - $3,700.
H19 Term and MX80Prt. - $800. Package Price - $4,300.
Doug Ramers (302)674-6386, 697-3444.

*****

HELP

Can you or any of your READERS HELP ME with this problem? I have a TRS-80 Color Computer 16K Extended Basic ROM 1.1 and an EPSON MX-80 with GRAFTRAX and interface board #8150 with 2K Buffer. I want to be able to draw a picture on the screen using a short program such as "POLYGON" or "JOYSTICK DRAW" and then print the result. Also, I have the ART GALLERY ROMPak and I would like to draw or create a picture and save it to tape and load it back in and print it. Is this possible? I have the cassette screen print program and the 8-Bit driver cassette that Radio Shack distributes but they have not given the results I wanted. Maybe I am not doing things in the proper order. I have only been able to print text so far on the EPSON, and I am sure it is capable of much more. I am beginning to think that I should have bought Radio Shack's line printer VII or VIII. They are supposedly able to dump a picture from the screen to the printer. ANY HELP you can give me will be appreciated.
Mike Davis, 6166 Char Mar Dr, Westerville, Oh 43081
(614)882-1954

*****

I am putting together a 6802 microcomputer system based upon the Percom SBC/9 and some MPI 8" drives. I need the following; disc controller card, minimal operating system (All I really want are block saves and loads), SS-50 RAM card, SS-50 ROM card, Editor/Assembler, high level control language.
Thomas Mason, 2402 Audubon Rd, Akron, Oh 44320
(216)836-7254

BUY DIRECT AND STUFF IT!

$19.50 EACH (Offer Expires 7/31/82)
- 30 Pin Extender Board
- Dual Serial I/O Board (30-pin)

$29.50 EACH (Offer Expires 7/31/82)
- 50 Pin Extender Board

All Extender boards fully labeled and shielded to prevent RF Interference.

$39.50 EACH (Offer Expires 7/31/82)
- 6809 CPU Board Baud Rate Generator
- Multiple I/O Board 2 ACIA and 2 PIA
- Video Display Board 6845 Based 80x24 Display
- Video Color Graphics Board 6847 Full Graph
- 5¼" Disc Controller Board (50 Pin)
- Modem Board Bell 103 Compatible, No hard to get parts!

$49.50
- 64K Dynamic Ram Board
- Completely Solder Masked Both Sides
- Full Documentation
- Reliable Phase I Refresh

$69.50
- Mother Board
- .093 Thick
- Baud Rate Generator on Board
- Full Address Decoding
- 4 or 16 Address Per I/O Slot

- All Boards Available Fully Stuffed
- VISA & Master Card Accepted
- Add $3.50 for shipping & handling
- All Boards Solder Masked Both Sides
- All Silk Screened Nomenclature
- Full Documentation Included

You're Right

We can't sell boards at this price for very long. But what better way to introduce you to DATA SYSTEMS "68". Order before July 31, 1982 (call 305-425-6800 for credit card orders) and realize savings of up to 50%. We guarantee your satisfaction.

DS 68

Data Systems "68"
2316 Diversified Way
Orlando, Florida 32804
(305) 425-6800

Limited Warranty

All DS68 products are guaranteed for a period of 90 days to be free of defects in materials and workmanship or return to Data Systems "68" for free replacement.

TERMS: Check, MasterCard, Visa. Sorry, NO C.O.D.'s. Florida residents add 5% sales tax. Specify type board(s) and quantity of each when ordering.

Now Comes

Computer Systems Center
The Folks Who Brought You DYNAMITE

DYNACALC™

The Electronic Spread-Sheet for 6809 FLEX

CSC

Enhance your computer's productivity with a powerful software tool for planning, manipulating data, and probing alternatives. With **DYNACALC,** you don't have to be a programmer. You simply enter your data and equations onto a giant "electronic spread sheet", and the answers appear as if by magic. Then play "What if. . . ?"—Change a data point or two, and watch the new answers appear almost instantly!

Until **DYNACALC** appeared, FLEX users have had to take a back seat to those with personal computers and the neat "VisiCalc" and "SuperCalc" programs that are available. But no longer. CSC has spent the thousands of programmer-hours required to bring the terrific capabilities of programs like these to the FLEX users of the world. **DYNACALC** is available now. Read the "System Requirements" note below to make sure your FLEX system is compatible (most are). Then see your local **DYNACALC** dealer or send us your order directly.

DYNACALC includes full 16-digit arithmetic for scientific and financial calculations. Some of the functions included are SIN, COS, TAN, LOG (and their inverses), X^Y, SQRT, INT, ROUND, ABS, MIN, MAX, SUM, AVERAGE, COUNT, LOOKUP, and NPV (net present value). **DYNACALC** uses super-fast 6809 machine code to guarantee its high performance. And of course **DYNACALC** talks to your existing printer.

The price is low for this type of computing power—just $200 (postpaid in U.S., Canada, and Mexico—others add $10 for air postage and handling) per copy. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size (5" or 8").

System Requirements:

6809 FLEX system with at least 32k total RAM, 56k preferred. Most any CRT terminal with directly addressable cursor and at least 80 columns across the screen. There is just one version of **DYNACALC,** and it is supplied with overlay customizers for various popular terminals, such as SWTPC CT-82, 8209, 8212, 8212W, Hazeltine 1420 and 1500 series, ADDS Viewpoint, Heath H19, Micro-term ACT-IV-V and MIME, TeleVideo, and others. Even if your terminal is one we have never heard of, it is usable simply by typing in the numbers required to make your terminal do the things **DYNACALC** needs it to do: cursor addressing, clear line and clear screen from cursor position, and (optional) dual intensity display and audible alarm.

DYNAMITE™

"THE CODE BUSTER"
DISASSEMBLES 6800 & 6809 MACHINE CODE INTO BEAUTIFUL SOURCE

- Convert your 6800 programs to 6809!
- Automatic LABEL generation
- Allows specifying FCB's, FCC's, FDB's, etc.
- Constants input from DISK or CONSOLE
- Automatically uses system variable NAMES
- DISK-to-DISK or DISK-to-CONSOLE operation

Includes 5" or 8" FLEX 9 diskette with relocatable object code. Full operating instructions (you'll learn in minutes!)

Order your DYNAMITE™ today
Only $60.00 postpaid in U.S. MC & VISA accepted
6809, FLEX 9, and 24K total RAM required

FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.
VisiCalc is a trademark of Personal Software, Inc.
SuperCalc is a trademark of Sorcim Corporation.
DYNACALC and **DYNAMITE** are trademarks of Computer Systems Center
Dealer inquiries invited.

13461 Olive Blvd. Chesterfield, MO 63017 314-576-5020

UniFLEX Version Available Soon!

INFOMAG DBMS

A data base management system specifically designed for microprocessor based computer systems.

- You specify new databases and screen entry masks.
- Define and work with subgroups of the master database.
- Sort groups.
- Audit trail can be maintained.
- Password protection.
- Control code definition for different terminals and printers.
- Columnar reports, modular reports, reports drawing from primary and secondary files.
- Statistics reports.
- Database files can be accessed by userwritten BASIC programs.

FLEX - $295.00 UniFLEX -$395.00

OSBORNE BUSINESS PROGRAMS

This ENHANCED IMPLEMENTATION of the Osborne and Associates Business Programs is the only implementation available with the full capability of the original Wang Minicomputer version.

FEATURES INCLUDE:

- KEYED FILES to eliminate slow searches and sorts.
- PASSWORD and MASTER PASSWORD PROTECTION to limit unauthorized access to your business data.
- LINKED can be run linked to G/L.

ACCOUNTS RECEIVABLE This package is an open invoice system, it prints aging reports, monthly statements, open item listings, etc. You can maintain trade discount schedules and scheduled penalties for late payment. FLEX.....$295 UniFLEX.....$395

ACCOUNTS PAYABLE This package is an invoice-linked system, it prints aging reports and will print out your checks. FLEX.....$295 UniFLEX.....$395

GENERAL LEDGER Accepts postings to the various accounts from external sources: Accounts Payable, Accounts Receivable, and Cash Journal. The normal posting is double entry to reduce off-balance posting errors. FLEX.....$295 UniFLEX.....$395

6809 SOFTWARE TOOLS

6800 RRMAC (Relocatable Recursive Macro Assemblers)
These are true Macro assemblers with external linkage capability. Variables may be local, global, or external. One assembler produces relocateable 6800 code which is loaded with the included Linking Loader. The second assembler, a cross assembler runs on your 6800 and produces relocatable 6809 object code from existing 6800 or new 6809 source files. Two assemblers plus a Linking Loader are including in this package.

FLEX.....$150

6809 RRMAC (Relocatable Recursive Macro Assemblers)
One assembler produces relocatable 6809 code which is loaded with the included link editor. The second 6809 assembler, a cross assembler, runs on your 6809 and produces relocatable 6800 code.

FLEX.....$150

SUPER SLEUTH DISASSEMBLER, enables the user to examine and/or modify object program files on disk or in memory on 6800/1/9 systems.

FLEX.....$99

GREAT PLAINS COMPUTER CO.

P.O. BOX 916 / IDAHO FALLS, ID 83402 / PH: (208) 529-3210
Visa and MasterCard accepted.

HOME CONTROL

HCOM software provides TOTAL control of ALL 256 possible devices. We provide the software and an adapter cable, You provide the RADIO SHACK PLUG N'POWER(tm) controller (026-1102) and a parallel port. This real time software may be adapted for whatever CPU speed you may be running and allows parameters to be modified for noisy electrical environments. Remember that NO ultrasonic links are required. Two NEW commands have been added to the standard six commands allowing you to LOCK and UNLOCK all devices under your control.

*AVAILABLE FOR PERCOM DB9 NOW!!
*AVAILABLE FOR SMIX 68 CONTROLLER W/DB9 SOON!!

5" DISK WITH ADAPTER & DOCUMENTATION-------$35.00

DPB DUAL PORT SERIAL BOARD

COMPATIBLE WITH BOTH SS30 & SS30-C
FULL RTS, DCD, CTS etc. JUMPER CONTROL
ALL CHIPS SOCKETED--USES 1488/1489 DRIVERS
ALL SUPPLY VOLTAGES REGULATED (+5, +12, -12)
TWO DB25 CONNECTORS ON BOARD FOR EASY HOOKUP

COMPLETE & TESTED WITH DOCUMENTATION-------$69.00

TG-1 BIT RATE GENERATOR

9 STANDARD BIT RATES FROM 110-19.2K SIMULTANEOUSLY
USES NO PARTS FROM YOUR PRESENT SYSTEM
FACILITATES UPGRADING TO 6850C STATUS

COMPLETE & TESTED WITH DOCUMENTATION-------$49.00

GOLD PLATED CONNECTORS ARE STANDARD

PLUG N' POWER is a trademark of RADIO SHACK.
DB9 is a trademark of MICROWARE SYSTEMS CORP.
SMIX is a trademark of SMOKY INC.

TEXAS COMPUTER

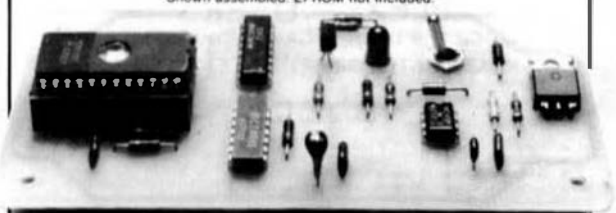
add for shipping $3.00

P.O. BOX 120818 ARLINGTON, TX 76012

VISA/MC 817-275-1848

EPROM PROGRAMMER

Shown assembled. EPROM not included.



For single supply 2516, 2716 & 2758 EPROMs. Connects through a user supplied interface to any computer system. Interfacing requires two 8-bit ports plus hand-shake lines. One of the ports must be software controllable for input or output. Timing is done via hardware, thus is indapendent of MPU clock rate. Verify erased. Program — entire or partial. Auto verify after programming. Transfer contents to RAM for modifying or duplicating.

Select Documentation for:	Interface to:
6502	6820 PIA or 6522 VIA
6800	6820 PIA
6809	6820 PIA
8080/8085/Z80	8255 PPI

Comprehensive documentation booklet contains schematic, instructions for construction, check-out and use, and a well commented assembly listing for the specified MPU.

Complete kit of parts (includes ZIF socket) ..... $ 45.00
Bare PC board and Documentation ..... $ 25.00
Software listings for additional MPUs (with purchase of Kit or PC board) ..... $ 5.00

Ordering: Specify MPU. Add 5% for P&H. Overseas add 10%. Ariz. residents add 5% tax.

Micro Technical Products, Inc.
814 W. Keating Ave., Dept. J
Mesa, Arizona 85202 • 602-839-8902

ALFORD'S GOOD NEWS!

UNLIMITED COMPUTER SPEECH...

Our speech synthesizers allow your computer to talk, LITERALLY! Using our VS-1 (for the SS-50 bus) or the SP-1 (for the COLOR COMPUTER), your programs can now communicate with speech! You can add a new dimension to almost any application with our synthesizers. Business programs, computer assisted instruction, industrial control, alarm systems, most anything can benefit from the power of speech.

We don't just sell you a hunk of hardware, either. Our synthesizers come with well-documented manuals detailing speech theory, programming examples, sample dictionaries, coding tables and more. We even include software to help to integrate these products into your system! Our unique "VOX-EDITOR" speech editing program, for instance, allows you to create and edit speech messages simply and easily. You can save the speeches created into files which can be directly read by Basic or other programs.

Both units come completely assembled and tested. No electronics knowledge or assembly is required. The SP-1 plugs directly into the ROM-PACK slot on the COLOR COMPUTER. Audio is fed back into the computer and routed to your television monitor. The VS-1 occupies one slot on the 30-pin I/O bus, and only requires the attachment of an external speaker (not supplied) for operation.

The VS-1 software is available for SSB DOS68 and all versions of 6800 FLEX (except mini-FLEX), and for SSB DOS69, FLEX-09 and OS-9, and may be ordered on 5- or 8-inch disk. The SP-1 software comes on a single cassette (disk versions not available at this time).

TEXT TO SPEECH TRANSLATOR...

We are now offering a speech translator for the VS-1 and SP-1 synthesizers.

The SC-1/VS-1 translator allows you to generate speech files from the keyboard, or from standard disk text files. The resulting file may be "fine-tuned" with the VOX-EDITOR, thus greatly reducing the amount of time needed to generate custom messages. Also, the SC-1/VS-1 translator allows you to simply speak your text files, allowing you to listen to your work while you do something else, eliminating the need to keep your eyes "glued" to the screen. You may also type from the keyboard directly to the translator.

The SC-1/SP-1 translator allows you to operate directly from Color Basic by adding a new port to Basic. Now, by typing "PRINT #1, "THIS IS A TEST", your computer will say "THIS IS A TEST"! By using control codes (PRINT #1,CHR$(xx)), you can make the translator pronounce everything which goes to the screen!

Flags are included in both versions which will cause the translator to pronounce punctuations. In this way, you can listen to every "jot and tittle" as it is read!

We plan to offer regular updates at a reduced price, so that as we continue to improve the translational accuracy of this product, you can take advantage of the updates without having to pay the full price!

The SC-1/VS-1 version is available for 6800 SSB DOS68, versions 5.1 and 6.0, and for all versions of FLEX (except mini-FLEX), 6809 versions are available for DOS69, FLEX-09 and OS-9. All may be ordered on 5-inch or 8-inch disk. The Color Computer version requires 16k or 32k memory (Extended Color Basic not required), and is supplied on cassette only.

The disk versions of the SC-1 translator include the source files, so you can make custom modifications to allow you to use it with other speech boards (we admit they exist!), or to change I/O addresses, etc... The 6809 disk version is self-relocating to place it at the top of available memory and out of the way of your other programs!

PUBLIC NOTICE...

FLEX is a trademark of Technical Systems Consultants, Inc. OS-9 is a trademark of Microware Systems Corporation. Radio Shack and TRS-80 Color Computer are trademarks of the Tandy Corporation. SSB ain't so picky! (love you, Rick, Larry, Jim, Debbie, Ken, etc...!). SCREDITOR III (no, Ed, we aren't gonna change it!) is a trademark of Alford & Associates, but then who else would use the name? For that matter, who can pronounce it?

TREK...

Some of you might not know it, but many of the games which you have seen written in Basic are emulations of games which were developed with compilers or assemblers for large machines. Some are pretty engrossing games, but all share two very bad problems ... question-and-answer play, and low speed. We went in the other direction. We took one of the most popular Basic games of all time, re-wrote it completely in assembly language, using full interrupt-driven, multi-tasking structure, and came up with what we feel is one of the best games around today!

Although our TREK uses alphanumerics exclusively in its display (thus making it compatible with most every memory-mapped display around), the lack of hi-res graphics is not missed due to the edge-of-the-seat action! The enemy space destroyers use multiple battle plans as they pursue you around the quadrants. Torpedoes destroy anything in their paths, forcing you to take continual evasive action. Damage control reports flash on and off as you take hits and repairs are made. Energy runs down continuously, making it impossible to sit back and relax. The only way to believe that so much action could be fitted on one screen is to play it!

TREK-68/6800 is available for 6800 processors running SSB DOS68 or any version of FLEX (except mini-Flex), and may be obtained on 5- or 8-inch disk. TREK-68/6809 is available for 6809 processors running SSB DOS69 or FLEX-09, and can be shipped on 5- or 8-inch disk.

COLOR-TREK is available on cassette for the TRS-80 COLOR COMPUTER, and requires 16K or more memory (Extended Color Basic NOT required). COLOR-TREK+ is identical to COLOR-TREK, except that it requires our SP-1 Speaker Pack voice synthesizer for operation, as the computer uses english speech messages continuously throughout program play.

TREK-68 requires a MEMORY-MAPPED display for operation as well as a source of interrupts (such as the MP-T timer) programmed for 50-120 interrupts per second. Also, a serial or parallel keyboard port is required (will not run with the MP-C interface).

FANTASTIC PROOFREADER...

We are now offering Peter Stark's SPELL'N FIX proofreading program for both FLEX and SSB! We think that this program is the best spelling checker/corrector available today, and is definitely the fastest and most flexible such program that we have used.

SPELL'N FIX provides a number of features that we have not seen on any other such program ... features such as multiple dictionaries, compressed dictionaries (25000 words on a single 5-inch diskette!), automatic dictionary updates, the option of simply printing errors, marking them in the file being proofread, or correcting them! Sector limits may be set to limit the length of file on each disk, text files may be split up, the dictionary split, and best of all, SPELL'N FIX doesn't let you down with DISK FULL errors! When the disk fills up, SPELL'N FIX automatically closes the output file and asks for another disk on which to continue!

In addition, you can perform most FLEX commands directly from SPELL'N FIX. The SSB version includes a directory list command which allows you to list disk files directly from SPELL'N FIX. This means that you don't have to stop and restart simply because you have forgotten a file name in the course of proofreading!

SPELL'N FIX is available for SSB DOS68 versions 5.1 and 6.0, and DOS69 (all versions), FLEX-9, 6800 FLEX (all versions except mini-FLEX), and can be ordered on 5- or 8-inch disk. Versions for mini-FLEX and PERCOM available directly from STAR-KITS. OS-9 versions will be available soon.

ALFORD & ASSOCIATES
P.O. BOX 6683
RICHMOND, VA., 23230
804-320-6722

SSB UTILITIES...

We are now shipping our UTILITIES #1 disk for DOS68 and DOS69 users. If your disk library is getting out of hand, or if you are going bonkers trying to make sense out of fifty, one hundred, or more files on a disk, this set of programs is for you!

We have developed a method to write a unique I.D. sector on SSB disks. In this sector, we include an 8-character disk access code, a 16-character disk name, an 8-character disk I.D., an 8-character disk creation date, an 8-character disk update date, and a 64-character comment field. Our NAME utility prompts you for the information and builds the information sector.

UPDATE then allows you to change the I.D. sector selectively. You can, without the access code, change only the update date and comments field, allowing you to make a quick change to reflect, say, the most recent change to the files on the disk. Using the ACCESS code, you can change everything except the disk creation date.

LOCK allows you to list- delete- and write-protect your files. With list protection, the list-locked files only list when you supply the disk access code!

LIST is a multi-option program, allowing you to list transients, non-transients, locked files (if you supply the disk access code), free space, disk information, or any combination of these.

PURGE is a quick and easy mass file deletion program. Each file on the disk is presented for your inspection. Multiple prompts then determine whether or not to delete each file.

TITLE generates BIG LETTER banner pages for your printed listings. You supply a name for the banner page of up to twelve characters, and TITLE generates the banner page with the current date and time (as contained in YDATE and YTIME).

DUMP is a formatted hexidecimal and ASCII memory dump. When directed to a hard copy device, page boundaries are honored to make dumps more readable.

COUNT allows you to quickly determine file sizes in lines and characters. Multiple files may be specified to allow you to count in a single pass those jobs which you have spread over several files.

UTILITIES #1 is available for DOS68, versions 5.1 and 6.0, and for DOS69 (all versions), and may be ordered on 5- or 8-inch disk. The manual also includes some very good information on the chained sector technique which SSB uses on their disks.

PRICES...

TREK-68/6800 disk, manual	$ 30.00
TREK-68/6809 disk, manual	$ 30.00
TREK-68 manual only	$ 12.00
COLOR-TREK cassette, manual	$ 17.00
COLOR-TREK+ cassette, manual	$ 24.00
COLOR-TREK manual only	$ 6.00
SPELL'N FIX disk, manual	$ 60.00
UTILITIES #1 disk, manual	$ 20.00
UTILITIES #1 manual only	$ 12.00
VS-1 complete, disk, manual	$194.00
VS-1 complete, disk, manual (less SC-01A)	$134.00
VS-1 disk, manual (use VOX-EDIT with brand X)	$ 30.00
VS-1 manual only	$ 10.00
SC-01A synthesizer chip	$ 45.00
SP-1 pack, cassette, manual	$170.00
SP-1 manual only	$ 14.00
SC-1/6800 disk, manual	$ 30.00
SC-1/6809 disk, manual	$ 30.00
SC-1/COLOR cassette, manual	$ 70.00
SCREDITOR III disk, manual, tutorial cassette	$100.00
SCREDITOR III manual and cassette only	$ 70.00

THE BEST NEWS...

God is truly a God of miracles. We could not begin to enumerate the miracles which He has performed for us, and wish to give thanks for His love and care in our lives. Without Him, we would not be in business. We hope that our concern and devotion to our customers reflects our love for Him. We do make mistakes, and occasionally an order is loused up or lost, but we'll do everything we can to correct any mistakes that we make. Thank you for your patronage.

SCREDITOR III DOES IT ALL...

This ad is being prepared using the SCREDITOR III word processor. Unlike any other editing package available for 6800/6809 machines, SCREDITOR III supports true multiple column editing capability. This ad was typed a column at a time. No cut and paste was used to prepare it (well, one, actually... our printer only prints 132 columns wide!).

SCREDITOR III is a true screen editor. What you are reading now is what we saw as we were entering the text! With full cursor movement capability, automatic justification, true multi-column insert, delete, paragraph splitting and welding, and edit capability of up to 251 character positions per line, SCREDITOR III makes document preparation a breeze! Also, you can define page footers and headers, automatically, delete and then re-insert them, or change them at any time.

You can merge multiple files with SCREDITOR III. You can even specify where to insert a file, what line in the file to start from, and how many lines of the file to use! You can extract or copy lines from the current file into another file. You can even do form letter printing with automatic line item merging for custom mailings.

Automatic justification (left, center, right, or full line) can be turned on or off at any time. Margins can be moved at any time. Just about any job can be done better with SCREDITOR III.

You can define "symbols" with SCREDITOR III. Up to 251 keystrokes may be included in a single symbol, including all edit and line commands, and later recalled with a single keystroke. Up to 26 symbols may be defined, and the symbols may be saved to a disk file for later recall.

Unlimited tabs may be set with SCREDITOR III, and may be changed at any time as you desire. Margins and tabs are continuously displayed for quick reference, along with all major edit modes, current column, current line, total lines, current display columns, etc...

SCREDITOR III runs with practically any SERIAL TERMINAL or with most any MEMORY MAPPED DISPLAY. If you change from a serial terminal to a memory-mapped display, or if you simply change display or terminal types, you don't have to buy a new copy of SCREDITOR III. Just run SYSGEN and define your new hardware! Even key definitions may be changed by SYSGEN! Terminals with special function keys may also be accommodated. Simply run SYSGEN, and hit each key as it asks you to, and SCREDITOR III will use that key in your system!

SCREDITOR III is designed to provide capabilities for any office, personal, programming, or other editing and formatting job that you might have. The applications are essentially unlimited.

The manual provides extremely detailed information about every mode, feature and function of SCREDITOR III, and was written with office personnel and the unsophisticated user in mind. A unique feature of SCREDITOR III is a tutorial cassette which makes learning to use SCREDITOR III simpler than any other such program on the market. The cassette takes you, step-by-step, through every function of SCREDITOR III, giving you practical examples and learning sessions with test files. No more reading for three weeks to learn how to do it!

SCREDITOR III is available to run under DOS68 versions 5.1 and 6.0 and all versions of 6800 FLEX (except mini-FLEX), and for DOS69, FLEX-9 and OS-9. SCREDITOR III may be ordered on 5- or 8- inch disk.

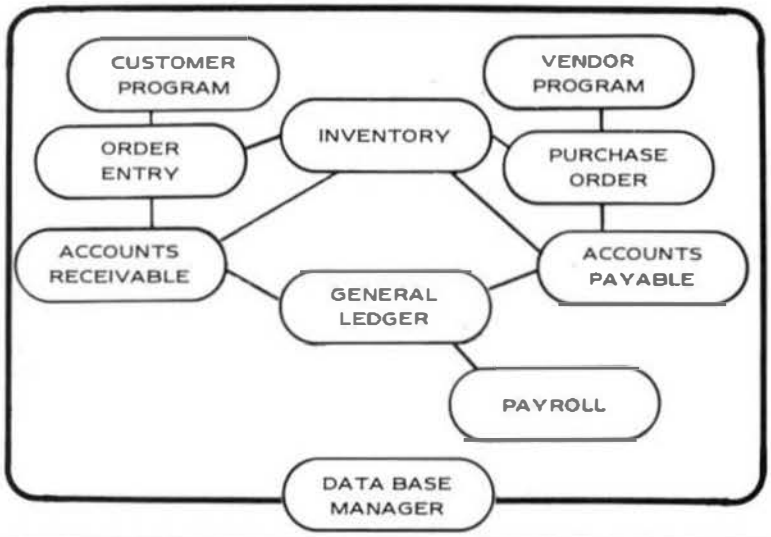
GENERAL INFORMATION...

Unless specified, all programs (except Color Computer) shipped on 5-inch disk for FLEX-9. Virginia orders add 4% sales tax. We ship by UPS on all domestic orders unless otherwise specified. COD's MUST go by UPS. Overseas orders shipped by air parcel post. On North American orders under $100, add $5 for shipping and handling. We pay shipping on orders over $100. On overseas orders, regardless of order value, add $10 for shipping. ORDERS RECEIVED WITHOUT SHIPPING WILL BE RETURNED UNFILLED.

We accept MASTERCHARGE, VISA and personal checks. Personal checks require two to three weeks to clear before processing can proceed. All checks must be in U.S. funds drawn on U.S. banks. Dealer and OEM pricing and licensing agreements are available. Please CALL for details. Open accounts to Fortune 1000's, schools, established accounts only. We regret this policy, but can no longer finance other businesses.

FLEX / UniFLEX SOFTWARE

THE TOTAL SYSTEM



STAND ALONE SOFTWARE:

- VENDOR PROGRAM
- PURCHASE ORDER
- ACCTS. PAYABLE
- CUSTOMER PROGRAM
- ORDER ENTRY
- ACCTS. RECEIVABLE
- PAYROLL
- GENERAL LEDGER
- INVENTORY (Retail)
- INVENTORY (Manufacturer)
- M.F.I. (Extensive Mfg./P.O.)
- BULK MAILINGS
- MAINTENANCE SCHEDULE
- DATA BASE MGR.
- DATA BASE (Utilities)
- SINGLE ENTRY LEDGER
- CHECK BALANCER
- COSTING & ESTIMATING
- CHURCH CONTRIBUTION PKG.
- CHURCH/SCHOOL BILLING
- FUND RAISING
- CUSTOM PROGRAMMING

INQUIRE ABOUT OUR COLOR COMPUTER SOFTWARE

SOFTWARE AVAILABLE THROUGH THESE DEALERS

Barney & Associates 113 W. Fifth Pittsburg, Kansas
Bonelli Media Assoc. 1116 N. 38th St. Allentown, Pa. 18104
Date Comp. P.O. Box 794 Hixson, Tn. 37343
Digitel Services 1509 Tieton Dr. Yakima, We. 98902
RAM Associates 49 Overlook Dr. Framingham, Mass. 01701
MRLectronics B.V. Buitendofdreef 280 2625 Re delft. The Netherlands
New Technology Systems P.O. Box 2 Cenetlan. DGO Mexico
Paris Radio 7 A Burton St. Darlinghurst. NSW Australia
Downeast Data Box 416 Harborside Rd. Northeast Harbor, Me. 04662
SWTP 38 Dover St. London, England

2457 Wehrle Drive • Buffalo, New York 14221 • (716) 631-3011

**FLEX AND UNIFLEX ARE TRADEMARKS OF TECHNICAL SYSTEMS CONSULTANTS.*

UNIVERSAL DATA RESEARCH INC.

64K SS-50 STATIC RAM

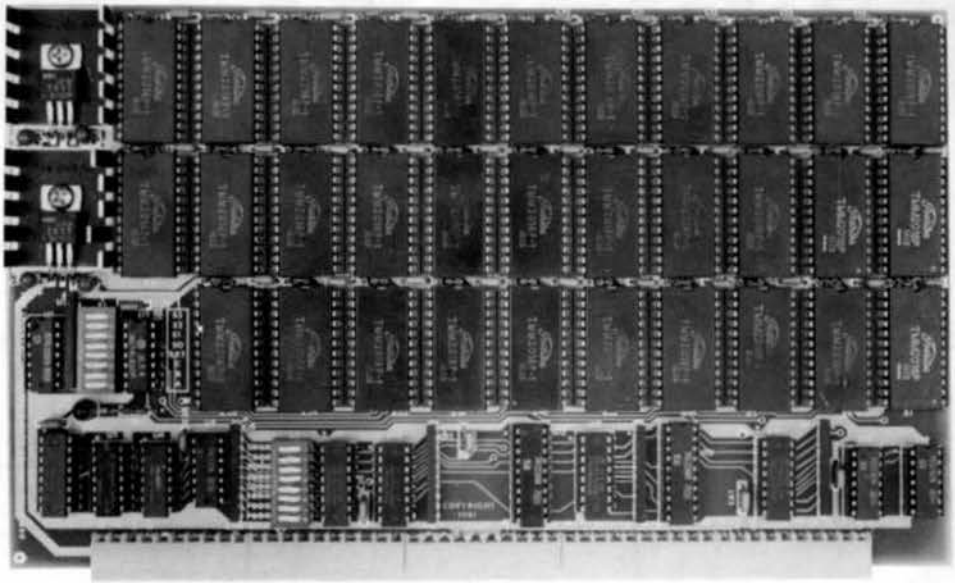
$299.00 (48K KIT)

NEW!

NEW!

LOW POWER!

RAM OR EPROM!



BLANK PC BOARD WITH DOCUMENTATION $52

SUPPORT ICs + CAPS -	$18.00
FULL SOCKET SET -	$15.00

ASSEMBLED AND TESTED **ADD $40**

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K Kit	$349
64K Kit	$395

16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Order under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

ELEKTRA COMPUTER PRODUCTS

ELEKTRA CABINET

Made of heavyweight 0.090" thick aluminum.
Interior Size: 18-1/2" wide by 21-7/8" deep x 6-3/4" high
Heavy duty A.C. line cord.
A.C. fuse holder
EMI filter
Fan with filter
Back panel has 10 cutouts for "D" type data connectors
Front panel has key on/off power switch, 2 illuminated push buttons (Reset and NMI/Abort), and 2 cutouts for 5-1/4" floppy drives.

Without power supply: $260.00
With 110v power supply, disk regulator board or two filler plates: $460.00
With 220v power supply, disk regulator board or two filler plates: $510.00

POWER SUPPLY

Higher quality linear power supply conservatively rated at 15a @ 8v, 3a @ 16v, 3a @ -16v
3 primary inputs to adjust for light, medium, and heavy loading

110v version: $175.00 220v version: $223.00

DISK REGULATOR BOARD with cables

Powers 2 5-1/4" floppy drives, $50

Filter (fan): $10.00

ELEKTRA CPU 8/9

Choice of 6802, 6808 or 6809 CPU
(6802 and 6808 are software compatible with the 6800 at the opcode level).

DEVICE		6809 ADDRESS	6802-6808 ADDRESS
3 2716 Eproms	Eprom #3	F800-FFFF	F800-FFFF and E000-E7FF
	Eprom #2	F000-F7FF	F000-F7FF
	Eprom #1	E800-EFFF	E800-EFFF
1K Scratchpad RAM		E400-E7FF	A400-A7FF and A000-A3FF
MC6840 Triple Timer		E210-E217	8280-8287

MC14411 Baud Rate Generator producing baud rates of:
Low Range 110, 150, 300, 600, 1200, 4800, and 9600
High Range 440, 600, 1200, 2400, 4800, 19200, and 38400
The board does not contain a DAT and does not support extended addressing.
The board supports DMA by either HALT or BUSREQ when a 6809 CPU is used.
DMA to the devices on the CPU card is not supported.
The board will run any of the MIKBUG™ compatible monitors in the 6802-6808 mode and SBUG-E, HUMBUG, and GIMIBUG-09 in the 6809 mode. The ELECTRA CPU 8/9 will run any of the popular disk controller boards with the appropriate software. Special versions of OS-9™ L1 are available.

Bare board: $30.00* Kit: $225.00* Assembled: $275.00

ELECTRA DPS Dual Port Serial Card

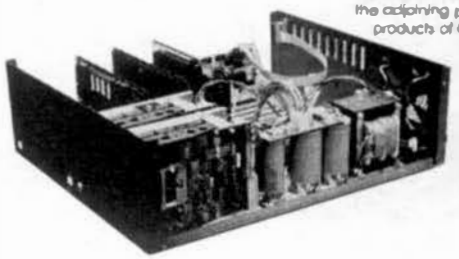
Fits the standard 30 pin SS-50 bus I/O slot
Can be configured for 4 addresses per port with the B port 2 addresses higher than the A port or for 16 addresses per port with the B port 4 addresses higher than the A port.
Each port is terminated at two 16 pin dip sockets, one socket configured for modem and the other socket configured for terminal or printer. RTS, CTS, DTR, DCD, DTR are appropriately implemented.
Each port has independent selection of baud rate.
Each port allows the interrupt request to be jumpered to the IRQ or FIRQ/NMI bus line.

Bare board: $20.00* Kit: $60.00* Assembled: $80.00
Assembled cable (two required for each interface board) $20.00 each

ELEKTRA DPP Dual Port Parallel Card

Fits the standard 30 pin SS-50 bus I/O slot.
Can be configured for 4 addresses per port or 16 addresses per port (occupying the first four addresses of the I/O slot).
The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals.
The interrupt request lines for each port may be individually jumpered to the IRQ or FIRQ/NMI bus line.

Bare board: $20.00* Kit: $60.00* Assembled: $80.00
Assembled cable (two required for each interface board) $20.00 each



The CPU, 56K memory board, and DMA controller board in the adjoining picture are products of GIMIX, Inc.

ELEKTRA Motherboard

Heavyweight 0.125" thick, 18" long by 9" wide.
11 memory (50 pin) slots. 4 or 8 slots may be cut off for shortening to 14" and 10" lengths respectively.
8 I/O (30) pin slots.
1" spacing between all memory and I/O slots.
On board baud rate generator with high and low ranges providing jumper selectable rates for each of the five I/O baud rate lines.
Complete address decoding and selectability for the I/O ports.
Choice of 4 8, or 16 addresses per I/O port.
Slow device circuitry permitting 1 MHz 30 pin disk controllers to run with 2 MHz 50 pin CPU boards.
Extended addressing capability for meeting SS-50C bus specifications.

Bare board: $80.00 Kit: $240.00** Assembled: $300.00**

**Gold square pin connectors instead of tin add $80.00

ELEKTRA Chassis

Includes cabinet, power supply, disk regulator board and power cable, and assembled motherboard with gold connectors, totally installed. Ready to use with documentation. $825 Add $50.00 for 220V.

GIMIX 16K memory board with control registers $195.00
GIMIX #15 memory board with 32K static RAM $295.00

*WARNING

AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled kits. AAA Chicago Computer Center does warranty and maintain service for our assembled boards. The customer should carefully take into consideration the small differential separating our kit and assembled prices when making his choice of purchase.

We have introduced our line of computer equipment with the purpose of offering the highest quality of components possible at affordable prices. These products are intended for OEM applications where it is the responsibility of the purchaser to integrate these components with suitable memory, disk controller(s), drives, and software along with I/O terminal(s) to form working computer system(s).

AAA Chicago Computer Center

120 Chestnut Lane • Wheeling, IL 60090 • (312) 459-0450

Closed June 26 thru July 18

Phone consultation available most weekdays from 4 PM to 6 PM

SMOOTH™ Software

SUPER MODEM PROGRAM

Transmit manually to distant computer

Transmit disk files (text) of any length to distant computer

Receive and save disk files (text) of any length on local disk system. If sending computer does not support an X-on/X-off protocol, then the received files are limited in size by the computer memory.

Tested to transmit and receive text at speeds up to 9600 baud. (CRT terminal must be capable of operating at a baud rate higher than the one the modem is operated at.)

Half duplex option in case distant computer doesn't echo

Echo option so user can simulate a time sharing system. (Super Modem Program doesn't support auto-answer but the source is provided for those individuals who wish to adapt our program to their special needs.)

Replaces CR with CR/LF (user option) for those using time sharing systems that don't transmit LF's

Slow disk file transmit (user option) based on character verification for use on time sharing systems to which disk files cannot be sent at speed suggested by the baud rate

Please specify 6800 SSB, 6800 FLEX™, or 6809 FLEX™ 5" or 8"
Manual and disk with both source and object code $75.00

STANDARD MODEM PROGRAM

Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option, nor X-on/X-off option. Reception of disk files is limited to those small enough to completely fit within the receiving buffer.

Please specify 6800 SSB, 6800 FLEX™, or 6809 FLEX™, 5" or 8"
Manual with instructions, source listing, and flow chart; disk with both source and object code $45.00

ALL IN ONE

Editor - Text Processor - Mailing Labels
Mailing Lists - Use any CRT terminal and printer

Supports Editing commands such as bottom, change, delete, find, insert (single line) input (multiple lines), list, next, overlay (with cursor editing, character deletion and insertion), overstrike (for selected darker text), print, start, set, top, underline, up, and verify

Supports Text Processing commands such as block copy, block move, centering, margin justification (widen and narrow), paging, and tabbing

Mailing Lists and Labels. Use the same mailing list disk file (with protected areas) for both mailing labels and repeat letters. Repeat letters are personally addressed to each person or selected persons on the mailing list.

Most Powerful File Handler found in any editor. Append the text file to the end of another or insert (merge) one file into another as designated by the line pointer. Print specified lines to your printer or to a disk file. Edit files larger than the text buffer. Does not produce output files when not desired. Delete disk files from the editor.

Printer commands. Control characters can be sent to the printer for format control either directly from the control terminal or by imbedding them in the text. The set command contains interface initialization and character output routines to support the SWTPC MP-C interface as well as the standard serial and parallel interfaces. Jumpers are also provided to use supplied printer routines. User selects the port address (0 thru 7, A or B) thereby eliminating the need for the user to install printer software routines. Editor can be initialized for either 4 or 16 addresses per port.

Editor allows exiting to either the monitor or DOS and then reenter (Warm Start) without destroying previously prepared text in the buffer. The Restart command erases contents in the buffer without the user having to reload the Editor.

The Editor allows the user to toggle between full duplex (no echo) and half duplex (echo) as needed. It responds to commands in both upper and lower case and can be used to create assembler source code and Basic programs as well as text.

Specify 6800 SSB, 6800 FLEX™, 6809 FLEX™, 5" or 8"	50.00
Printed source listing is available for an additional	35.00

Software by Technical Systems Consultants, Inc.

Flex™ (includes Editor and Assembler)	150.00
UniFLEX™ (includes one year maintenance and update)	450.00
Editor	50.00
Assembler	50.00
68000 Cross Assembler on 6809	250.00
6809 Cross Assembler on 6800	100.00
Text Processor	75.00
Extended Basic	100.00
Basic recompiler (specify standard or extended)	50.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™) (Add $75.00 for one years maintenance and update)	225.00
Sort/Merge Package	75.00
6809 Flex™ Utilities	75.00
6800 Flex™ Utilities	100.00
Debug Package	75.00
Diagnostic Package	75.00

Software by Microware Systems Corp.	UPDATE	SOURCE	MANUAL	OBJECT
OS-9™ Level One Operating System	75.00	400.00	40.00	200.00
OS-9™ Level Two Operating System	75.00	N/A	40.00	500.00
BASIC09™	75.00	N/A	25.00	200.00
OS-9™ Macro Text Editor		300.00	15.00	125.00
OS-9™ Interactive Assembler		300.00	10.00	125.00
OS-9™ Interactive Debugger (Disk version)		100.00	10.00	50.00
C/S Cobol Compiler	250.00	N/A	80.00	900.00
Pascal Compiler	100.00	N/A	40.00	400.00

SWTPC	
	Kit / Assembled
DMF2 Disk Controller Board (NEW)	595.00
DMF2 Disk Controller Board (Used, very limited quantity)	450.00
SWTPC FLEX™ Disk and manual	35.00
SWTPC FLEX™ Disk without manual	10.00
DC-2 Disk Controller (Limited Quantity)	N/A / 125.00
4K RAM Board (Limited Quantity)	N/A / 40.00
SBUG-E (2716 compatible)	19.95
SWTBUG (6830 compatible, limited quantity)	N/A
MP-A2 6800 CPU Board	150.00
MP-S2 Serial interface (dual port)	N/A / 120.00
MP-LA Parallel interface (dual port, limited quantity)	40.00 / 60.00
MP-L2 Parallel interface (dual port)	N/A / 120.00
MP-R Single voltage 2716 prom programmer	N/A / 114.50
MP-N Calculator board	54.95 / 92.00
MP-T Interrupt timer	N/A / 92.00
MP-8M 8K 4044 Memory board (limited quantity of kits)	150.00 / 275.00
S32 Universal Static Memory Board	N/A / 124.50
MP-09 6809 CPU board	N/A / 295.00
68 Chassis, P.S., 68B09 CPU, 8K, RAM, One Serial Port	N/A / 799.00

Universal 68XX Bare Motherboard, 6800/6809, 4/16 addresses per port, 8 50 pin/8 30 pin slots, baud rate generator, 15 1/8" x 9 3/8" 65.00
ELEKTRA Motherboard (bare) 80.00

Connectors (10 pin, Titanium-Tin plated 5 microns for near gold quality)

Male with square cross section pins	each .50
Female	each .75
Gold, Male w/square pins or female	each 1.50

SMOOTH™ and **ELEKTRA™** are trademarks of AAA Chicago Computer Center
FLEX and **UniFLEX** are trademarks of Technical Systems Consultants, Inc.
OS-9 and **BASIC09** are trademarks of Microware Systems Corp.
GIMIX™ and **GHOST™** are registered trademarks of GIMIX Inc.

ELEKTRA™ SS50 Computer Products

ELEKTRA Dual drive cabinet for 5-1/4" drives with power supply, line cord, fuse, power switch, and power cable to drives	125.00
ELEKTRA Dual drive cabinet, power supply, ps cable for 8" drives	350.00
Cabinet for dual 8" drives only	250.00
Power supply for dual 8" drives only	120.00
PS cables only (Specify brand and type of 8" drives)	30.00

Special Software

4K 6809 HUMBUG	75.00
4K 6800 HUMBUG (RAM needed at $A000 and $D000)	6.00
2K 6800 HUMBUG (With cassette LOAD and PUNCH)	40.00
2K 6800 HUMBUG (Extra commands instead of cassette software)	40.00
Other HUMBUG versions including video versions are available	
Spell'n Fix by Peter Stark	89.29
Dynamite Disassembler	60.00
SUPER SLEUTH disassembler System	99.00

DISK DRIVES 30 day guarantee, single and double density capability

5-1/4", single head, 40 tracks	$250.00
5-1/4", single head, 40 tracks, flippy	260.00
5-1/4", double head, 40 x 2 tracks	335.00
5-1/4", single head, 80 tracks	335.00
5-1/4", single head, 80 tracks, flippy	345.00
5-1/4", double head, 80 X 2 tracks	470.00
MPI - Service Manual	20.00
Siemens Manual	10.00
8", single head, 77 tracks	375.00
8", single head, 77 tracks, Thin-Line	450.00
8", double head, 77 X 2 tracks	495.00
8", double head, 77 X 2 tracks, Thin-Line	525.00
Microlime 6800 Calendar and Clock Board (assembled and tested)	105.00
Bareboard, connector, and documentation only of above (See review Feb. 1980 68 Micro Journal)	35.00
Microlime II	75.00
Data Mart 16K E-ROM barebord 12708 chips)	30.00

Printers

Epson MX-80 (Centronics compatible parallel interface)	439.00
(with Serial RS-232 interface option)	add 75.00
Spare Print Head	39.95
Spare ribbon cartridge	15.00
C. Itoh Comet I 125 cps, 9 x 7, bidirectional, serial or parallel	295.00
Okidata Microline 82A 120 cps, 9 x 9, bidirectional, serial and parallel	495.00
Tractor for 82A	70.00
Optimal Technology, Inc. EP-2A-79 Eprom Programmer	169.00
(Personality Modules extra for above programmer)	
Optimal Technology, Inc. 30 pin parallel I/O board for EP-2 -79	35.00
Software package for EP-2A-79 (Specify 6800 or 6809)	29.95
EP-2A-78-M-01 Eprom Programmer (User configures and supplies power)	79.95

Smoke Signal Broadcasting

Monitor on 2716 with manual (Specify Cheltain or SWTPC)	75.00
OCB-4A Double Density Controller Board for 5" and 8" with DOS	549.00
OOS690 DOS Update with Editor and Assembler (Specify 5" or 8")	75.00
LMB-1A Motherboard	399.00
SCB-69 6809 CPU Board	399.00
M-16-X 16K Static Memory Board	195.00
M-24-X 24K Static Memory Board	295.00
M-32-X 32K Static Memory Board	395.00

GIMIX

2 MHz 6809 Plus CPU, time of day clock, battery backup, 1K NMOS RAM	578.05
CMOS RAM substitution	8.00
GIMIX Dynamic Address Translator	35.00
SWPTC comatible DAT	15.00
Missing cycle detect card	38.23

Disk Controllers (All have data seperators and can be used with either single or double headed drives)

5" single density controller without 1771 chip	158.38
5" single density controller complete	198.48
5" and 8" single density controller complete	226.58
5" double density controller with variable precomp	298.28
DMA 5" AND 8" double density controller with variable precomp	588.68
GIMIX version of FLEX™ (without Editor and Assembler)	90.00
Double disk regulator card	68.22
Ribbon cable for two 5 1/4" disk drives (short)	34.96
Ribbon cable for two 5 1/4" disk drives (long)	44.96
Ribbon cable for two 8" disk drives (long)	44.26
8" disk drive cabinet with power supply	848.18

Memory	CMOS WITH BAT. BACKUP	NMOS NO BAT. BACKUP
16K Static RAM Board with control registers**		185.00
32K Static RAM Board with 32K of RAM installed*		295.00
*discontinued, limited quantity available		
64K Static RAM Board with 24K of RAM installed	N/A	348.27
64K Static RAM Board with 32K of RAM installed	518.36	398.37
64K Static RAM Board with 48K of RAM installed	N/A	518.47
64K Static RAM Board with 56K of RAM installed	728.58	578.57
64K Static RAM Board with 64K of RAM installed	798.64	638.67
16 Socket EPROM/ROM/RAM Board		238.32
8K Promboard (2708)		98.34
4K PPD 4K Prom Board and 2708 Prom Burner		198.00

I/O Boards

Single port 30 pin serial interface (Requires 1 cable set)	88.41
Dual port 30 pin serial interface (Requires 2 cable sets)	128.43
8 port 50 pin serial interface with baud rate generator	318.46
Dual port 30 pin parallel interface (Requires 2 cable sets)	88.42
8 port 50 pin parallel interface with interrupt generator	198.45
Cable sets for above boards (specify board)	24.95
2MHz 6809 PLUS Computer System with 56K Memory*	2498.29
Above System with #58 Controller and Special Software Pkg.*	2988.59
Above System with #68 Controller and Special Software Pkg.*	3248.49
*with CMOS RAM and Battery Backup	add 300.00
M inframe (Chassis, PS, Switches, Fan, Motherboard, Baud Rate Gen.)	1198.19

Shipping and handling estimates:

Within the Continental U.S., please add 3% ($5.00 minimum)
Foreign, prepaid and add 10% ($10.00 minimum) for light items only
Heavy items must be prepaid and will be shipped Emery Air Freight Collect.

Please phone during consultation hours if questions arise regarding shipping fees.

Master Charge, Visa, and American Express honored

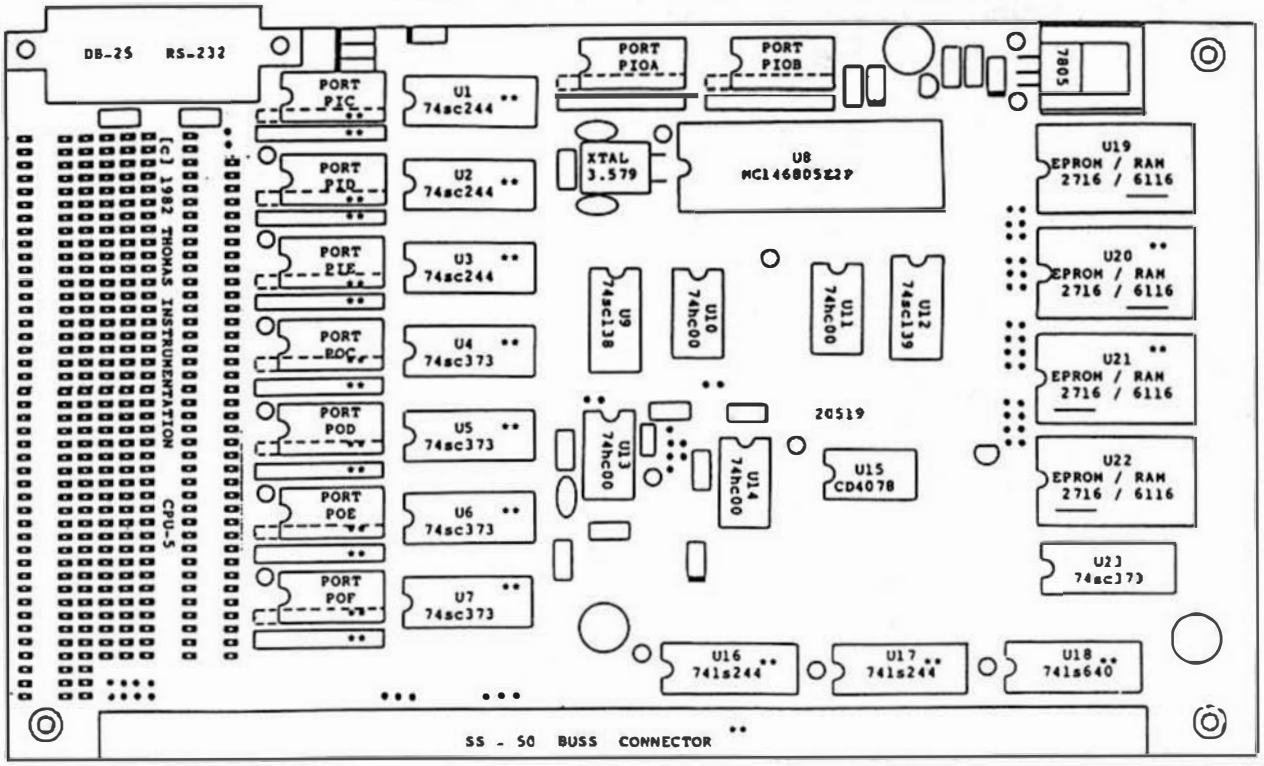
AAA Chicago Computer Center **Closed June 26 thru July 18**

120 Chestnut Lane, Wheeling, IL 60090
(312) 459-0450 Phone Consultation available most weekdays from 4 PM to 6 PM

Our apology. We are running 4 weeks behind answering our mail regarding technical questions. Please phone during consultation hours for technical help.

Prices and inventory are subject to change without advance notice.
This ad is our catalog

*** NEW 6805 ALL CMOS CARD ***



CPU-5-B	Blank card with documentation	$49.00
CPU-5-P	Partial card (parts marked ** not incl.)	$229.00 } Includes EPROM Monitor
CPU-5-F	Full card with all parts shown (fully assembled)	$349.00 } and 115 vac Power Pak.
CROSS-ASSEMBLERS	for FLEX ((c) TSC) 6809 system	$150.00

SOFTWARE

VDISK Treat extended memory as a super fast disk drive
6809 source & object $149.00
6809 object $ 99.00

OUTSIDE MODEM PROGRAM incl. source
UniFLEX version $100.00
FLEX version 6800 & 6809 $ 50.00
HAYES SMART MODEM $249.00

CROSS ASSEMBLERS for 6800,6801,6805 runs on 6809 FLEX $150.00

TV-EDIT Screen oriented Editor for FLEX 6809 $ 95.00
Please specify 5" or 8" Disk when placing order.

CHIPS

Please note when ordering IC's that a $100.00 Minimum is in effect Orders less than the minimum will be charged an additional $10.00 for handling

6116P-3	2MHZ	$13.50
2016P-2	2MHZ	$10.50
2716	1MHZ	$ 6.50
2716-1	2MHZ	$16.50
146805E2P		$25.00
68B02		$10.00
68B09		$25.00
68B21		$ 6.50
68B40		$12.00
68B50		$ 6.75
74LS640		$ 3.35

HARDWARE

NEW MEMORY PRICES TO INTRODUCE VDISK

S-R/R without memory chips	$120.00
with 8K NMOS@2MHZ	$169.00
with 16K NMOS@2MHZ	$199.00
with 32K NMOS@2MHZ	$299.00
with 48K NMOS@2MHZ	$399.00
Bare Card	$ 49.00

Extender Cards assemb. with logic aid

SS-50/50C	$ 35.00
SS-30	$ 25.00
SP-1 Prototype Board A/T	$195.00
SP-1 Bare Card	$ 49.00
SS-50 Wire Wrap Board B/C	$ 39.00
SS-30 Wire Wrap Board B/C	$ 20.00
6802 Super CPU A/T	$235.00
6802 Super CPU B/C	$ 59.00
Video RAM, B/C $49.00, A/T	$195.00
Parallel I/O, B/C $49.00, A/T	$139.00
SS-50 Backplanes 4,6,8,12,16 position @ $5.00 per slot w/o connectors	
SS-30 8 pos. BP w/o connect.	$ 39.00
Transition Card $49.00, A/T	$ 95.00
Molex Gold Male $1.60, female	$ 1.60
Molex Tin Male $ .40, female	$ .50

THOMAS INSTRUMENTATION

168 EIGHTH STREET — AVALON, N.J. 08202 (609) 967-4280
NJ RES. INCLUDE 5% SALES TAX
Allow 3 Weeks for Checks to Clear
CONT. USA INCLUDE $3.00 SHIPPING, CANADA $6.00, FOREIGN $12.00
MASTERCARD, VISA, and C.O.D. ACCEPTED

Send $20.00 to receive full documentation, schematics, & source listings for all boards currently in production.

'68' MICRO JOURNAL ADVERTISERS INDEX

This index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

SMOKE SIGNAL BROADCASTING®

31336 VIA COLINAS WESTLAKE VILLAGE, CA 91362 TEL (213) 889-9340 / TWX 910-494-4965

MICROPRODUCT BULLETIN

ARE YOU REALLY SATISFIED?

. . . with the **PRODUCTS** you are selling?
. . . with the **PROFITS** you are making?
. . . with the **SUPPORT** you receive?

Let's face it—

The PRODUCTS, the PROFITS, and the SUPPORT are the key factors in a Dealer-Manufacturer relationship.

IF YOU WANT INCREASED SALES, BIGGER PROFITS, AND REAL SUPPORT — CALL SMOKE SIGNAL BROADCASTING.

We know you've heard that story before. In fact, it probably compelled you to become a dealer in the first place! What you probably **didn't** know is this:

Too many computer companies make ambitious claims when you first express interest, but **very few of them will continue to hold up their end of the deal once you have signed up.**

Smoke Signal **will** deliver what is promised . . . plus much more!

Being a Smoke Signal Broadcasting Dealer is easy. we **don't** require large initial orders; we **don't** impose monthly quotas or contracts on you. What we **do** have are good, reliable, high-quality computer systems; excellent pricing and large discounts. Those factors alone provide enough dealer incentive **without** messy monthly quotas.

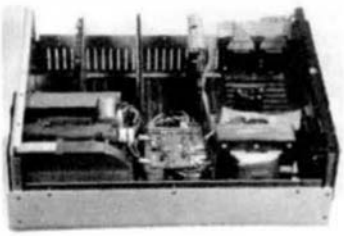
Our business computers, the CHIEFTAIN Series, are state-of-the-art 6800/6809 and SS-50 Bus. We run FLEX, OS-9 Multi-user, plus our own DOS68D and DOS69D, with absolutely no changes on your part. Our marketing plans include you, the Dealer — our **real** sales force.

Convince yourself. Call or write Deborah Conrad (213) 889-9340 for more Dealer and OEM information.

FLEX - OS-9 LEVEL ONE - UNIFLEX - OS-9 LEVEL TWO

ONLY GIMIX Systems can be configured to run any of these.

GIMIX systems utilize the most powerful 6809 operating systems: FLEX, UniFLEX, OS-9 LEVEL ONE and TWO -- the systems the PROs use. This means a wide selection of software to choose from as well the ability to develop sophisticated, multi-user/multi-tasking programs on your GIMIX System.



The GIMIX CLASSY CHASSIS™ consists of a heavy-weight aluminum mainframe cabinet which provides more than ample protection for the electronics and 1 or 2 optional 5¼" drives.

Backpanel connectors can be added for convenient connection of terminals, printers, drives and other peripherals.

A 3 position locking keyswitch enables users to disable the front panel reset button to prevent accidental or unauthorized tampering with the system.

The **GIMIX** system mother board provides fifteen 50 pin slots and eight 30 pin I/O slots -- the most room for expansion of any SS50 system available. The on board baud rate generator features 11 standard baud rates, 75 to 38.4K, for maximum versatility and compatibility with other systems. Extended address decoding allows the I/O block to be addressed anywhere in the 1 megabyte address space. All components feature Gold plated connectors for a lifetime of solid connections. All boards are fully buffered for maximum system expansion.

Each **GIMIX** Mainframe System is equipped with an industrial quality power supply featuring a ferro-resonant constant voltage transformer to insure against problems caused by adverse power input conditions such as A.C. line voltage fluctuations etc. The supply provides 8 volts at 30 amps and plus or minus 16 volts at 5 amps, more than enough capacity to power a fully loaded system and two internal drives.

The 2MHz GIMIX 6809 PLUS CPU board includes a time of day clock with battery back-up and 6840 programmable timer to provide the programmer with convenient, accurate time reference. Later addition of 9511 or 9512 arithmetic processors is provided for on the board. The unique **GIMIX** design enables software selection of either OS-9 or FLEX, both included in many complete GIMIX systems.

GIMIX STATIC RAM boards require no complicated refresh timing cycles or clocks for data retention. GIMIX memory boards are guaranteed for 2 MHz operation with no wait state or clock stretching required.

Our low power NMOS RAM requires less than 3/4 amp at 8V for a fully populated 64K board. For critical situations, our non-volatile 64K byte CMOS static RAM boards with built in battery back-up retain data even with system power removed. A fully charged battery will power this board for a minimum of 21 days. A write protect switch permits CMOS boards to be used for PROM/ROM emulation and software debugging.

The GIMIX DMA controller leaves the processor free to perform other tasks during disk transfers - an important feature for multi-user/multi-tasking systems where processor time allocation is critical. The DMA board will accomodate up to 4 drives 5¼" or 8" in any combination running single or double density single or double headed. Programmed I/O Disk Controllers are also available.

GIMIX systems are designed with ultimate **RELIABILITY** in mind. You can choose from the below featured systems or select from our wide variety of components to build a custom package to suit your needs.

GIMIX 2MHz 6809 System Including: CLASSY CHASSIS, 6809 PLUS CPU BOARD, 56KB STATIC RAM, 2 SERIAL PORTS W/CABLES, GMXBUG MONITOR, FLEX, and OS-9 LEVEL 1 ..... **$3248.49**

FOR TWO 5¼" 40 TRACK DSDD DRIVES ADD ..... **$ 900.00**

GIMIX 128KB WINCHESTER SYSTEM Including: CLASSY CHASSIS, 6809 PLUS CPU BOARD, 128KB STATIC RAM, 4 SERIAL PORTS W/CABLES, 5¼" 80 TRACK DSDD FLOPPY DISK DRIVE, 19MB 5¼" WINCHESTER HARD DISK, OS9 LEVEL 2, EDITOR AND ASSEMBLER **$8998.09**

50HZ Versions Available, 8" Drives Available — Contact GIMIX for Prices and Information.

The Sun Never Sets On A GIMIX!

GIMIX users are found on every continent, including Antarctica. A representative group of GIMIX users includes: **Government Research and Scientific Organizations** in Australia, Canada, U.K. and in the U.S.; NASA, Oak Ridge, White Plains, Fermilab, Argonne, Scripps, Sloan Kettering, Los Alamos National Labs, AURA, **Universities:** Carleton, Waterloo, Royal Military College, in Canada; Trier in Germany; and in the U.S.; Stanford, SUNY, Harvard, UCSD, Mississippi, Georgia Tech, **Industrial users** in Hong Kong, Malaysia, South Africa, Germany, Sweden, and in the U.S.; GTE, Becton Dickinson, American Hoechst, Monsanto, Allied, Honeywell, Perkin Elmer, Johnson Controls, Associated Press, Aydin, Newkirk Electric, Revere Sugar, HI-G/AMS Controls, Chevron, **Computer mainframe and peripheral manufacturers,** IBM, OKI, Computer Peripherals Inc., Qume, Floating Point Systems. **Software houses;** Microware, T.S.C., Lucidata, Norpak, Talbot, Stylo Systems, AAA, HHH, Frank Hogg Labs, Epstein Associates, Softwest, Dynasoft, Research Resources U.K., Microworks, Meta Lab, Computerized Business Systems.

GIMIX Inc. reserves the right to change pricing and product specifications at any time without further notice.

GIMIX® and GHOST® are registered trademarks of GIMIX Inc.
FLEX and UniFLEX are trademarks of Technical Systems Consultants Inc.
OS-9 is a trademark of Microware Inc.

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

The Company that delivers

Quality Electronic products since 1975.

E X T R A
FLEX & RS COLOR COMPUTER

With the EXATRON INTERFACE and DISK CONTROLLER

GOOD NEWS #1 — The popular, easy to use, and very powerful FLEX9 Disk Operating System has been running on the Radio Shack Color Computer for several months using the Special EXATRON 32K Interface and Disk Controller. This System adds 32K Memory to a 16K Color Computer, providing 48K of RAM for User Programs, the FLEX Disk Operating System, etc. This is accomplished by adding 16K more Memory between the 16K in the Color Computer and the BASIC ROM's, and adding 16K of Memory ABOVE the BASIC ROM's (which is REQUIRED for the FLEX Disk Operating System). This System requires NO MODIFICATIONS to the Computer. It operates as a Single Sided, SINGLE DENSITY, 35 Track, System, and controls up to 4 Disk Drives. Most MAJOR Programs must be modified to run on this System.

F-MATE(EX) — This is a set of SPECIAL SOFTWARE ROUTINES supplied on a 5 1/4" Diskette which "Installs" the Special General FLEX Operating System on the Radio Shack COLOR COMPUTER as a "bootable" System for normal use. The normal FLEX9 Utilities such as COPY, CATalog, LIST, RENAME, DELETE, etc. are included to provide a fully usable Disk Operating System.

PATCHES — To patch and make functional various standard TSC Utilities and programs including APPEND, ASMB, EDIT, PUTLDR, SAVE.LOW, XBASIC, and others, A special NEWDISK routine allows disks made on the COLOR COMPUTER to be read or written on other FLEX9 Systems.

- - - - SYSTEM REQUIREMENTS - - - -

FLEX9 Special General Version including the Editor and Assembler (NOTE: the Editor and Assembler each sell for $50.00, so you get FLEX9 for $50.00). $150.00

Special EXATRON 32K Expansion/Disk Controller $299.95

F-MATE(EX) FLEX9 Conversion for EXATRON SYS.
when purchased with Spec. FLEX9 Sys. $49.95
when purchased without Spec, FLEX9 Sys. $59.95

Screen-Clean - R.F. Noise Eliminator for EXATRON SYSTEM
Wired and Tested $39.95

Radio Shack 16K COLOR COMPUTER, with Extended BASIC and ready for the above items $499.95

- - - FOR USERS THAT ALREADY HAVE FLEX9 & Disk Drives - - -

Radio Shack DISK CONTROLLER with F-MATE(RS)
and a Special Two Drive DISK CABLE $289.95

- - - DISK DRIVE PACKAGES, with RS Controller - - -

These Packages include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable:
PAK #1 ==> 1 Single Sided, Double Density Sys. $529.95
PAK #2 ==> 2 Single Sided, Double Density Sys. $779.95
PAK #3 ==> 1 Double Sided, Double Density Sys. $649.95
PAK #4 ==> 2 Double Sided, Double Density Sys. $949.95

- - - - - - - PARTS AND PIECES - - - - - - -

Radio Shack Disk Controller	$199.95
1 ea. Single Sided, Double Density Disk Drive	$249.95
1 ea. Double Sided, Double Density Disk Drive	$349.95
Single Drive Cabinet with Power Supply	$79.95
Double Drive Cabinet with Power Supply	$99.95
Single Drive Disk Cable for RS Controller	$24.95
Double Drive Disk Cable for RS Controller	$34.95
Micro Tech. Prods., Inc. LOWER CASE ROM Adapter	$74.95
Radio Shack BASIC Version 1.1 ROM	$34.95

VISA or

MASTER CHARGE accepted

Add $25.00 Shipping & Handling For Complete Set

With the RADIO SHACK COLOR COMPUTER DISK CONTROLLER

GOOD NEWS #2 — The popular, easy to use, and very powerful FLEX9 Disk Operating System is AVAILABLE NOW for the Radio Shack COLOR COMPUTER with the Radio Shack COLOR COMPUTER DISK CONTROLLER. This system requires a Version 1.1 BASIC ROM and 64K RAM. This is easily accomplished on a nor el Radio Shack 32K Color Computer, which already has the Version 1.1 ROM and memory bank select jumpers, by replacing the existing RAM Chips with KNOWN GOOD 64K Chips and enebling one NOR gate. If you do not have a 32K System, you can have it updated by a Radio Shack Service Center, or purchase a Version 1.1 ROM and modify it yourself. Data Comp can supply GUARANTEED 64K Memory Chips and instructions for the modification (see below).

F-MATE(RS) — This is a set of SPECIAL SOFTWARE ROUTINES supplied on TWO 5 1/4" Disks which provide the conversion routines for developing a normal "bootable" FLEX9 System for operation WITH THE RADIO SHACK COLOR DISK CONTROLLER. The F-MATE(RS) "INSTALLATION" Disk contains the routines which "marry" the FLEX9 Disk Operating System to the specific requirements of the Radio Shack Color Computer. The F-MATE(RS) "UTILITIES" Disk contains the Special Routines developed for this Package and accomplish the normal Input/Output conversions, along with the necessary Software to activate the 64K Memory System, relocate the Display Screen Memory and variable areas, provide NEW, INDEPENDENT KEYBOARD and DISPLAY CAPABILITIES, etc.

FEATURES — Data-Comp's F-MATE(RS) DOES NOT REQUIRE A "PATCHES" CONVERSION. This adaptation allows ALL FLEX9 Compatible Software which uses the normal FLEX9 I/O routines to run on the Radio Shack COLOR COMPUTER WITHOUT MODIFICATION. Special COLOR COMPUTER Utilities supplied include:

1. FIVE different DISPLAY SCREENS (supplied with the Source Code so you can develop your own character set). 32 x 16 (the normal CC Screen), 32 x 24, 42 x 24, 51 x 24, and 64 x 24 Display Screens are available via a simple system command.
2. SAVE ROM's - a routine which allows saving the BASIC ROM's to a FLEX9 Disk, so normal Radio Shack BASIC can be called and run with the 64K Memory still enabled.
3. DISK and MEMORY Diagnostic Routines.
4. EXTENDED KEYBOARD including full "CONTROL" Key functions, an "ESCAPE" Key, and 12 user definable keys.
5. SPECIAL NEWDISK Routine for Formatting Single or Double SIDE, Single or Double DENSITY, 35, 40 or 80 TRACK Diskettes.
6. System capable of running up to THREE DOUBLE SIDED DRIVES, or FOUR DRIVES if none are Double Sided.

- - - - SYSTEM REQUIREMENTS - - - -

FLEX9 Special General Version w/ Editor and Assembler (the EDITOR and ASSEMBLER normally sell for $50.00 each, so you get Gen. FLEX for $50.00) $150.00

F-MATE(RS) FLEX9 Conversion for R. S. Disk System
when purchased with Spec. FLEX9 Sys. $49.95
when purchased without Spec. FLEX9 $59.95

Set of eight 64K RAM Chips w/ mod instructions $99.95

64K RAM Radio Shack COLOR COMPUTER System $649.95

- - - - - - SPECIAL SYSTEM PACKAGES - - - - - -

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, Special General Version of FLEX9 , F-MATE(RS) , and a Box of 10 Double Density Diskettes; a COMPLETE, ready to run SYSTEM on your Color TV Set $1379.95

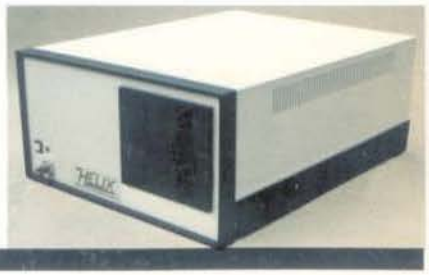
DATA-COMP
SOUTH EAST MEDIA
P.O. Box 794 Chattanooga TN 37443

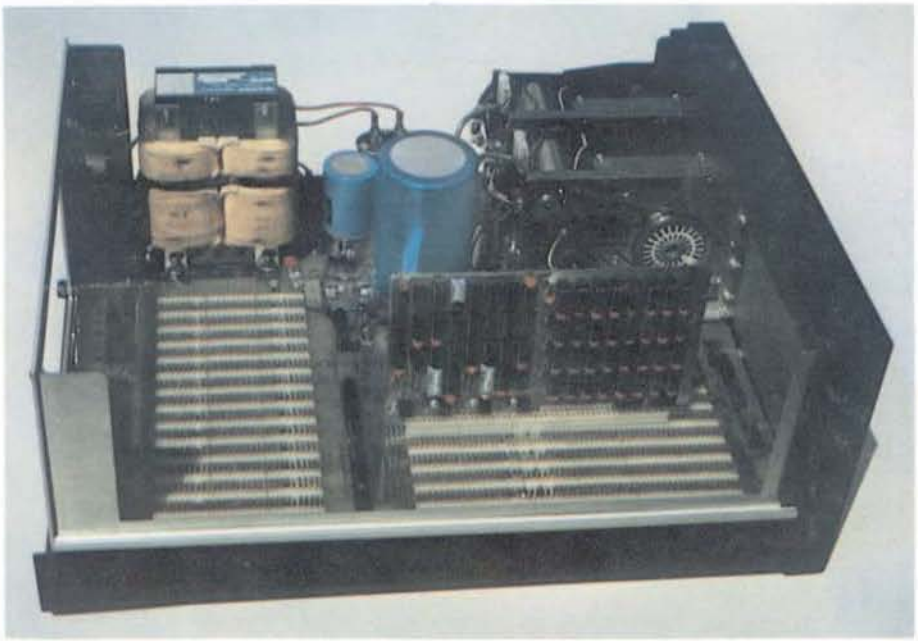
1-615-842-4601

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

HELIX™

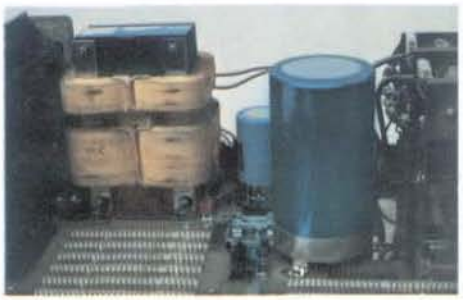




THE MAINFRAME
- Industry Standard Optima[1]™ Cabinet
- Largest Constant Voltage Power Supply in the Industry
- S-64 Bus gives 16 Bit Power and S-50 Bus Compatibility
- 10 Main (S-64) Slots
- 14 I/O (S-30) Slots plus 2 On-board
- On-board Baud Rate Generator to 38.4Kb
- Space and Power for two 5¼" Disk Drives
- Full Address Decoding for I/O Slots
- Two RS-232 Serial and Two parallel Ports On-board
- Single Board Construction for Reliability
- Faraday Shielded Bus Lines give "Text Book Clean" Signals

THE PROCESSORS
6809
- Standard 2 MHz Operation
- Standard DAT Compatible with GIMIX and SWTPC
- Standard 6840 Interval Timer
- Standard 1K Scratchpad RAM
- Standard Clock/Calendar with Battery
- Provision for Programmers Console

68000
- Standard 8 MHz Operation
- Memory Management Hardware
- Provision for Programmers Console
- 16 Bit Power and 8 Bit Compatibility



The HELIX™ computer system represents the latest advance in S-50 bus computer systems. Relying on the physical nature of S-50 bus connectors to guarantee compatibility, the HELIX adds 14 bus lines (becoming S-64) to allow a 68000 processor to operate with full 16 bit data transfer and 24 bit addressing, while at the same time providing full interchangability with existing S-50 components.

Offered with a selection of processors, memories, and peripheral controllers, a HELIX system can be configured for applications ranging from advanced hobbyist to multiterminal time-sharing.

Designed to offer the utmost in speed, reliability, and utility at a reasonable price, it represents a new standard of quality for those who require a professionally designed computer for professional use.

THE POWER SUPPLY
- Ferro-resonant Transformer for Line Noise and Under-Voltage Protection
- Conservative 25 Amps at 8.5 Volts
- Conservative 5 Amps at ±16 Volts
- Conservative Component Rating for Reliability

THE COMPONENTS
- Fully Socketed
- Gold Plated Bus Connectors
- Only "B" Series 68XX Components Used
- Only Top Grade Logic Circuits Used
- Industrial Grade Co ponents Throughout

THE MEMORIES
DM-64
- Field Proven
- Proprietary Memory Control Logic
- Fully Transparent Refresh
- Tested at 2.5 MHz Operation

DM-512
- 512K Bytes on a Single S-64 Board
- 16 Bit Power and 8 Bit Compatibility
- Runs in Existing S-50 Systems where Physical Space Allows
- Full 24 Bit Addressing
- Fully Transparent Refresh

THE PRICES
Because of the variety of configurations possible, full pricing cannot be given. Representative prices are:

- 64K 6809 HELIX $1995
- 64K 68000 HELIX $2595
- 512K 6809 HELIX $4450
- 512K 68000 HELIX $4995

HAZELWOOD COMPUTER SYSTEMS

7413 N. Lindbergh, Hazelwood, Missouri 63042 (314) 281-1055

Dealer and OEM Inquiries Invited. We support our Dealers.

Optima is a Trademark of Scientific-Atlanta